

**Introduction to
Information &
Communication
Technologies**
CL-1000

Lab 06
Web Basics: HTML and CSS

National University of Computer & Emerging Sciences – NUCES – Karachi



Contents

1. Front -End Development	2
2. What is HTML?	2
3. What is CSS?.....	2
4. What is JavaScript	2
2. Visual Studio Code	3
2.1 Installing Visual Studio Code on Windows.....	3
2.1 Installing Prettier Extension for VS Code.....	7
2.2 Installing Live Server Extension for VS Code	8
2.3 Prettier - Code Formatter	11
2.4 Live Server.....	11
4. Document Object Model (DOM)	11
5. HTML & CSS in Action	12
5.1 Image and Text Element.....	14
5.1.1 Image Element	16
5.1.2 Text Element.....	17
5.1.3 Add Styling Here comes CSS	18
6. Formatting Code	20
7. Basic HTML Boilerplate.....	24
8. Working with Text.....	25
8.1 Emphasis Tag	25
8.2 Styling the Emphasis Tag.....	26
9. Headings.....	27
9.1 Importance of Headings.....	27
9.1.1 Key Points Regarding the Heading Tags	29
10. None-Breaking Space in Paragraphs (HTML Entities).....	30
11. Hyperlinks	32
11.1 Key Points about Hyperlinks and the Anchor Tag.....	32

1. Front -End Development

Front-end web development is the process of building the part of a website that users directly interact with. It involves creating an engaging, intuitive interface where users can easily navigate and perform tasks. The primary goal of front-end development is to ensure the user experience is smooth, functional, and aesthetically pleasing. By combining three key technologies—**HTML**, **CSS**, and **JavaScript**—developers build the layout, design, and interactive elements of websites. These tools allow developers to craft everything from simple static pages to complex, dynamic applications that respond in real time to user input.

2. What is HTML?

HTML (Hyper Text Mark-up Language) is defined as the fundamental building block of the web, serving as a markup language that **structures content on web pages**. It allows developers to **create** and **organize text, images**, and other **multimedia elements**, facilitating the **design** and **layout** of a website. HTML provides the framework for web pages, enabling browsers to render the content correctly for users.



3. What is CSS?

CSS (Cascading Style Sheets) is defined as the language used for **styling web pages**. It allows developers to **control the appearance of HTML elements**, such as **layout, colors, fonts, and spacing**. By separating content from presentation, CSS **enhances the design flexibility** of web pages, making it easier to create visually appealing and **responsive sites**. CSS works by **applying styles** to **elements** defined in HTML, thereby determining how the content is displayed to users



4. What is JavaScript

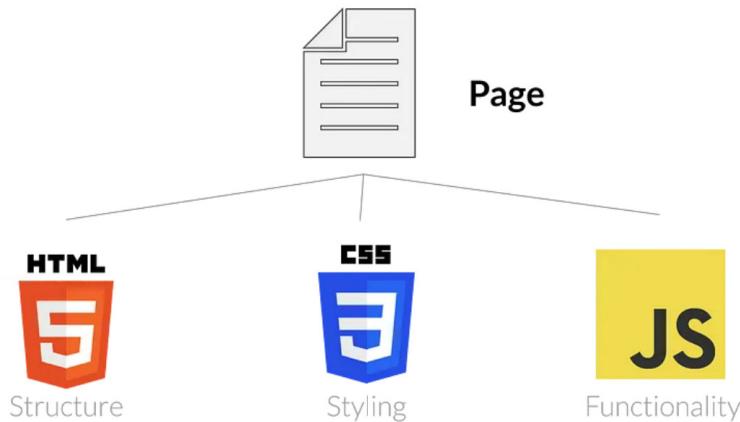
JavaScript fits into the web development stack alongside HTML and CSS by adding **interactivity** and **dynamic behavior** to web pages. While HTML provides the **structure** and **content** of a webpage, and CSS controls its **appearance** and **layout**, JavaScript is responsible for making the webpage **interactive**, allowing users to engage with the page by performing tasks like **form submissions, animations, or dynamic content updates without reloading the page**.



Together, they form the foundation of modern web development:

- **HTML:** Defines the content and structure.
- **CSS:** Styles and formats the appearance of the content.
- **JavaScript:** Adds interactivity and functionality.

JavaScript enables features such as **form validation**, **dynamic updates**, and **user-triggered events**, making it an essential part of creating interactive web experiences.



2. Visual Studio Code

Visual Studio Code, also commonly referred to as **VS Code**, is a source-code editor developed by **Microsoft** for **Windows**, **Linux**, **macOS** and **web browsers**. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded version control with Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add functionality.



We will use VS Code for running our Python programs throughout this lab.

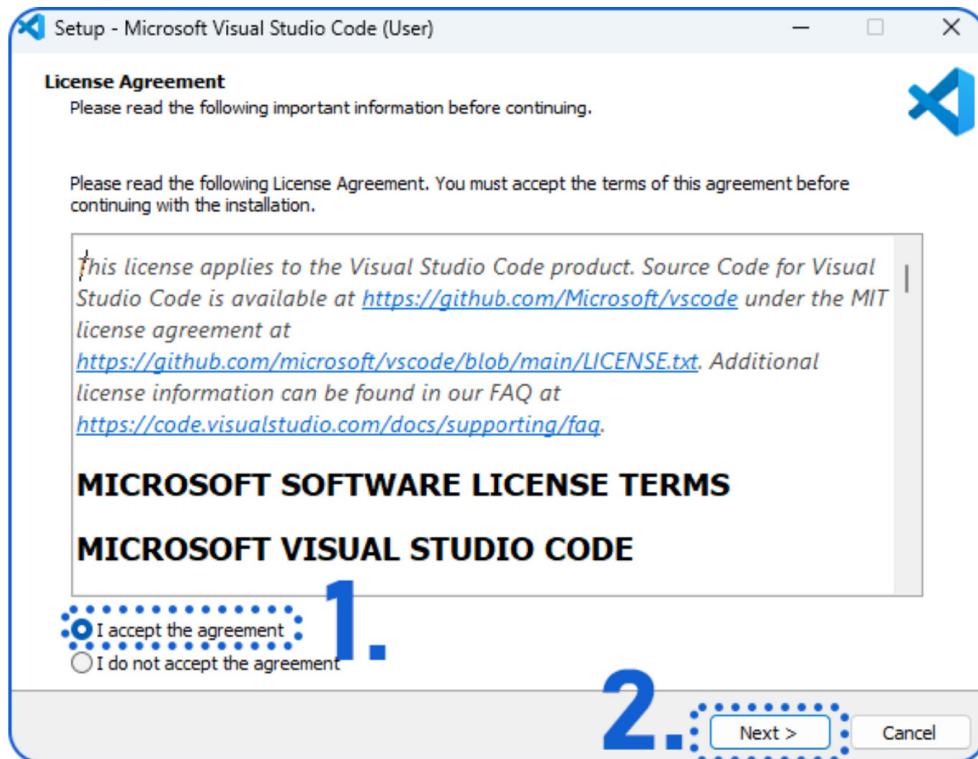
2.1 Installing Visual Studio Code on Windows

1. Click on the following link to download the Visual Studio Code installer.

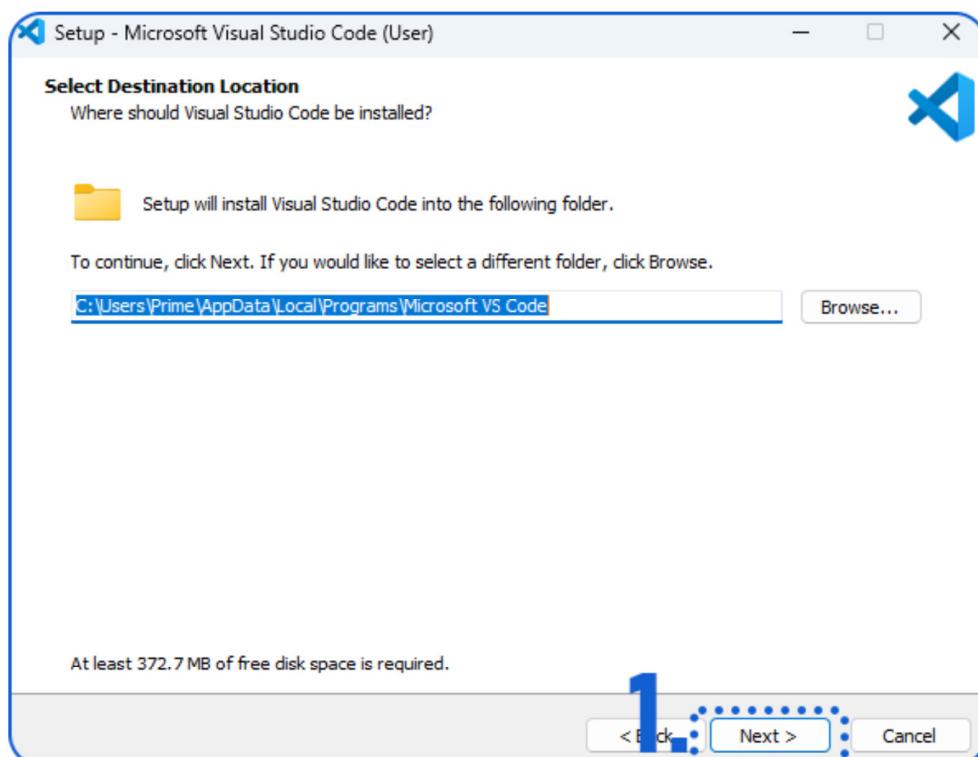
Link: [VS Code](https://code.visualstudio.com)

A screenshot of a web browser displaying the Visual Studio Code website at <https://code.visualstudio.com>. The page features a large "Code Editing. Redefined." heading with a "Download for Windows" button. On the right, there's a preview of the VS Code interface showing an Explorer sidebar with files like "button.ts", "button.css", and "grid", and a central code editor window with some TypeScript code. A message at the bottom of the code editor says "Create a new button component" with "Accept" and "Discard" buttons. The status bar at the bottom shows "Changed 9 lines".

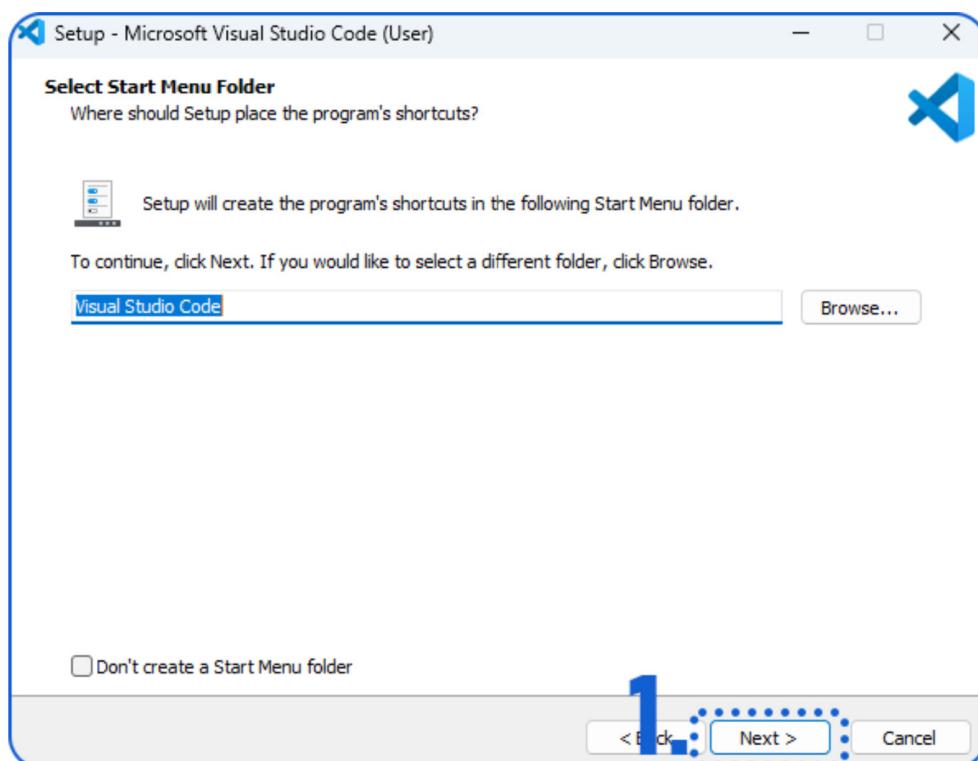
2. Run the downloaded installer.
3. Accept the License Agreement.
4. Click on **Next >**.



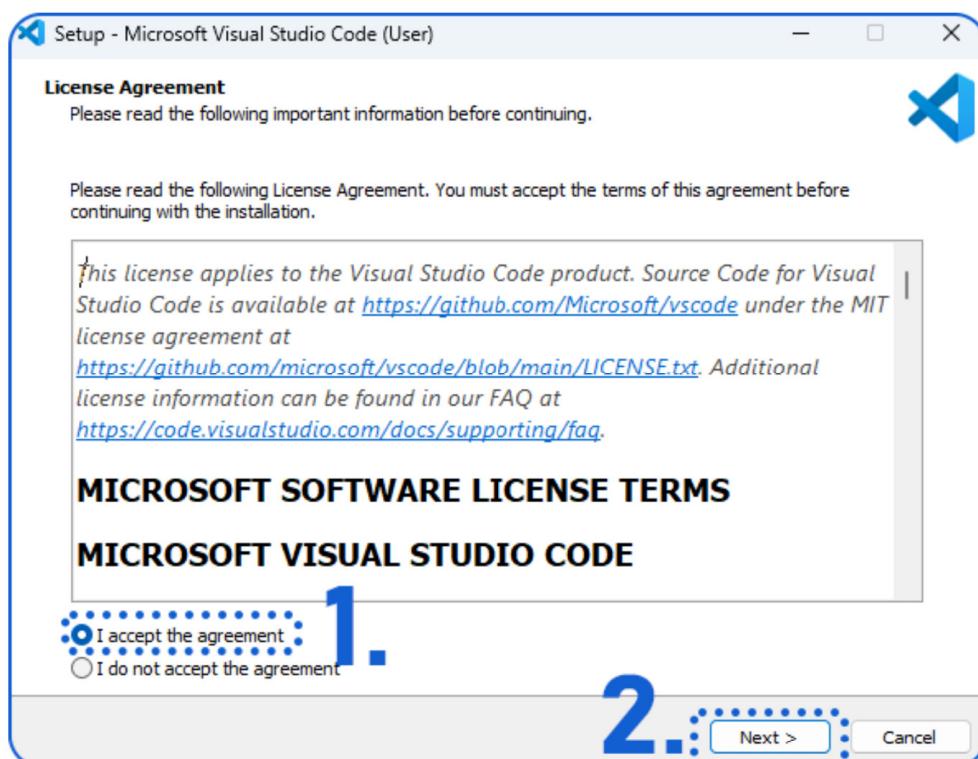
5. Leave the installation directory to default and click on **Next >**.



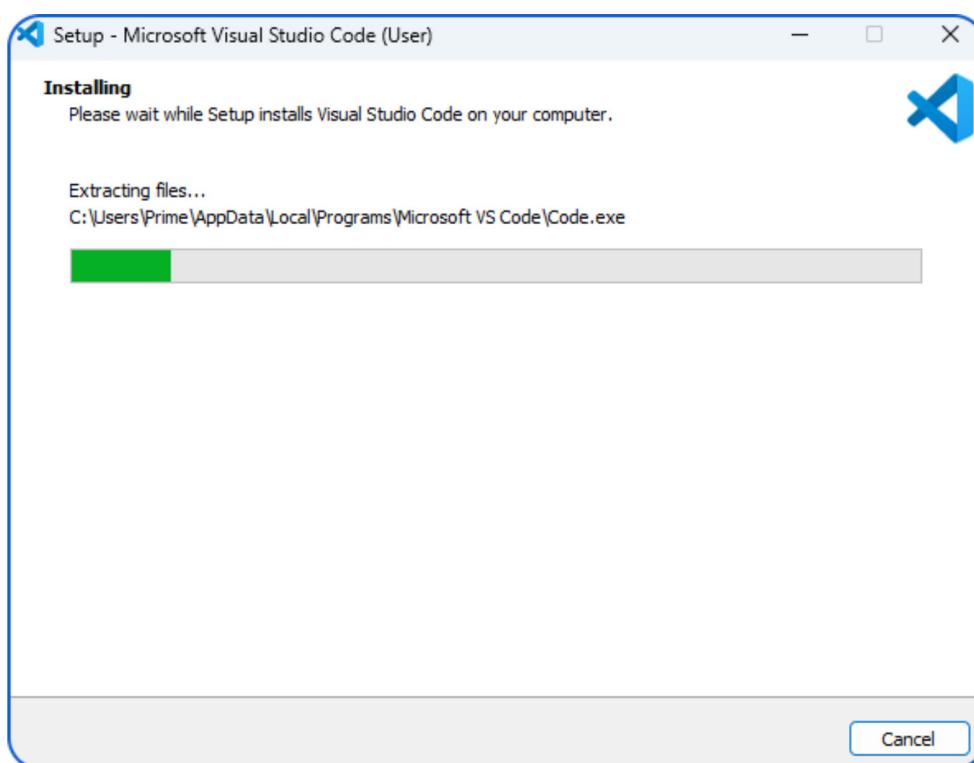
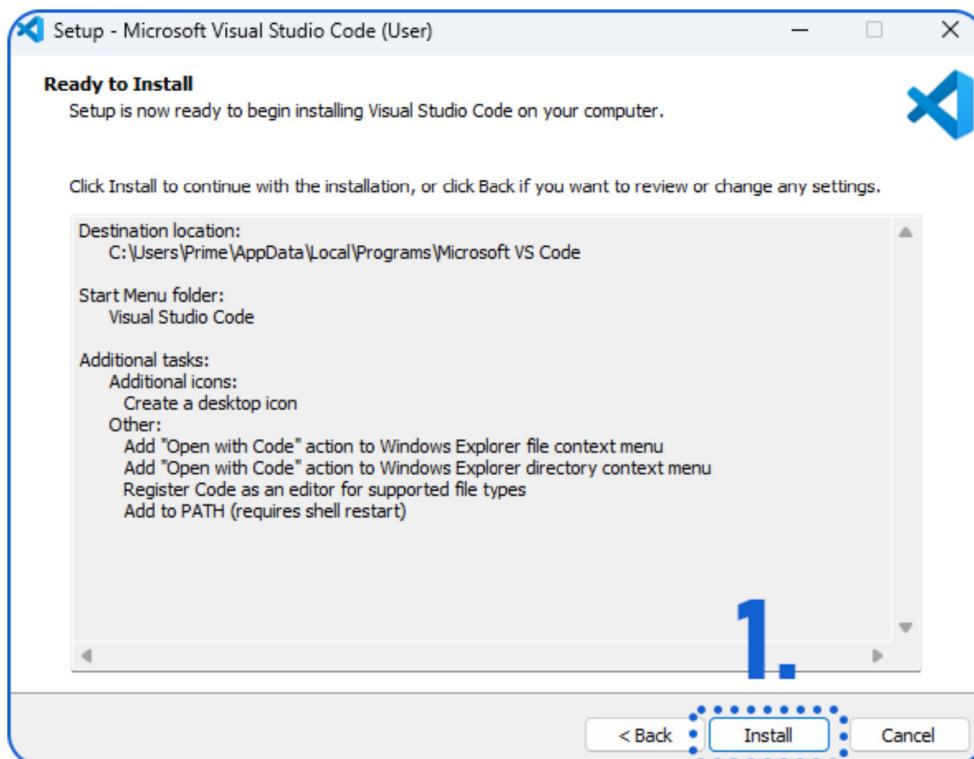
6. Click on **Next >**.



7. Check all the options and click on **Next >**.

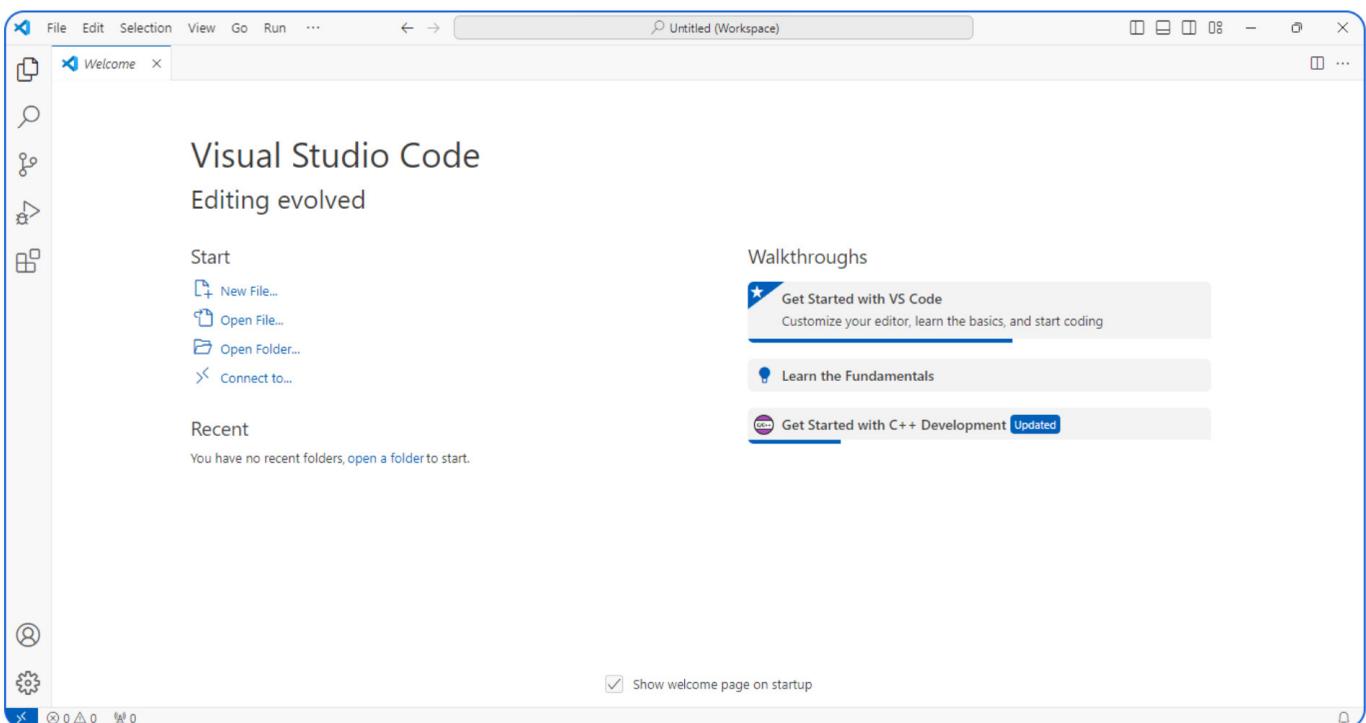


8. Click on Install.

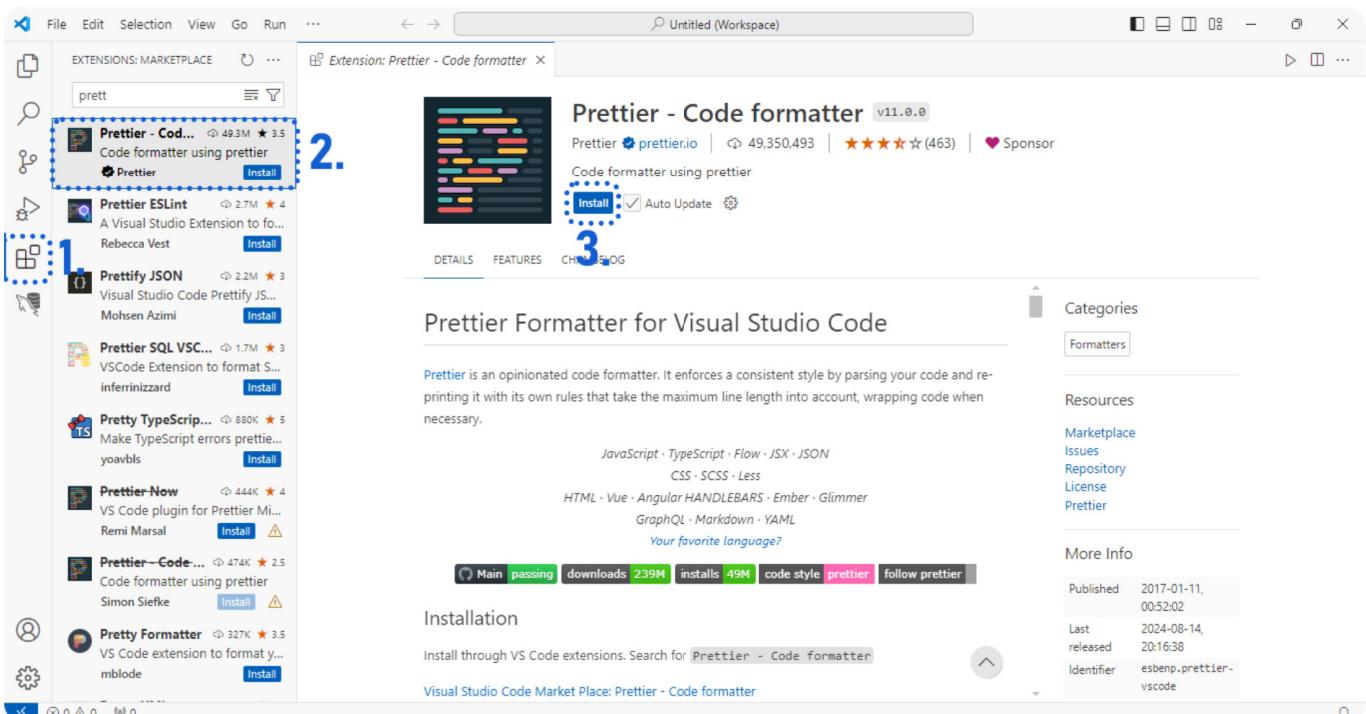


2.1 Installing Prettier Extension for VS Code

1. Open Visual Studio Code.



2. Click on the **Extension: Marketplace** icon at the left-most pane.
3. Type **Prettier** and click on the **Prettier** extension to install it.
4. The extension tab will open, click on the **install** button. Wait for the installation to finish.

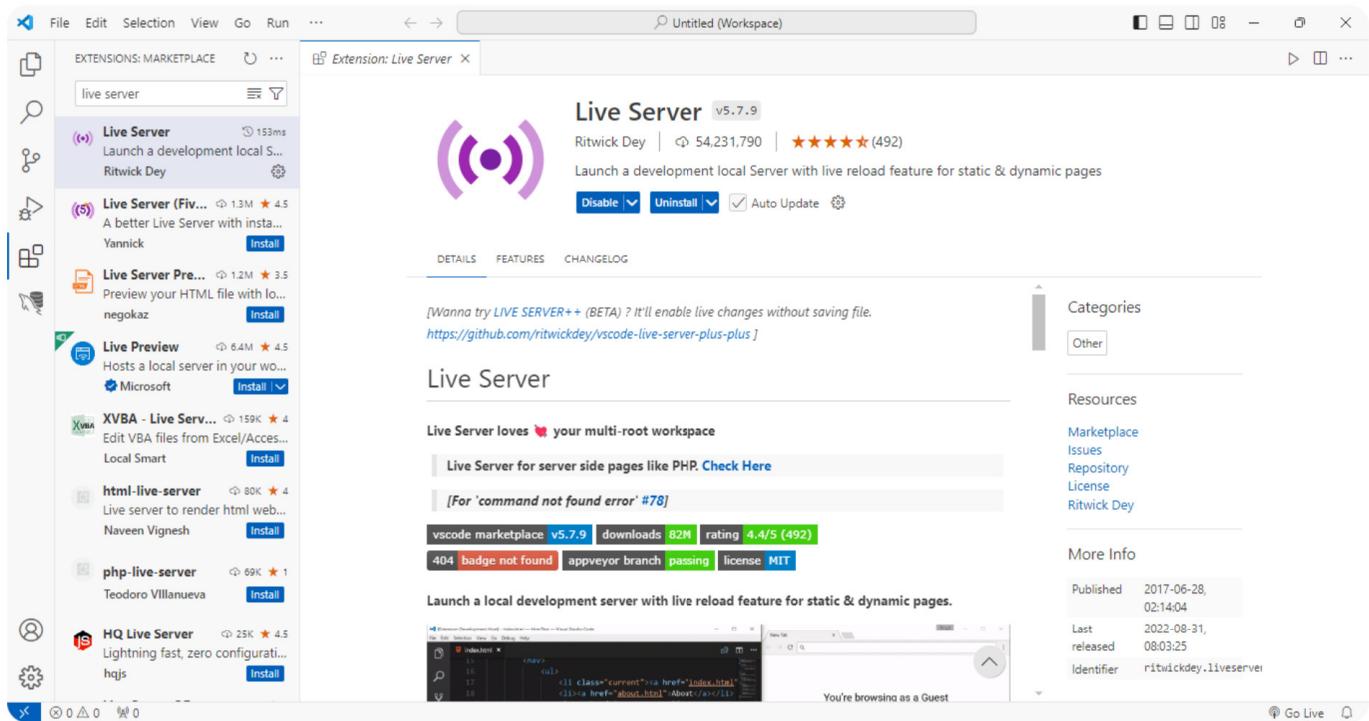


The screenshot shows the Visual Studio Code Marketplace interface. On the left, there's a sidebar with various icons for file operations like Open, Save, Find, and Run. The main area has a search bar at the top with the text 'Untitled (Workspace)'. Below it, a list of extensions is shown, with 'Prettier' selected. The right side displays the details for the 'Prettier - Code formatter' extension, version v11.0.0. It includes a large icon of a colorful abstract pattern, the extension name, its rating (4.9 stars), and download count (49,350,493). Below this, there are tabs for 'DETAILS', 'FEATURES', and 'CHANGELOG'. A sidebar on the right lists categories like 'Formatters' under 'Categories' and 'Marketplace', 'Issues', 'Repository', 'License', and 'Prettier' under 'Resources'. At the bottom, there's a 'More Info' section with details such as publication date (2017-01-11), last release date (2024-08-14), and identifier (esbenp.prettier-vscode).

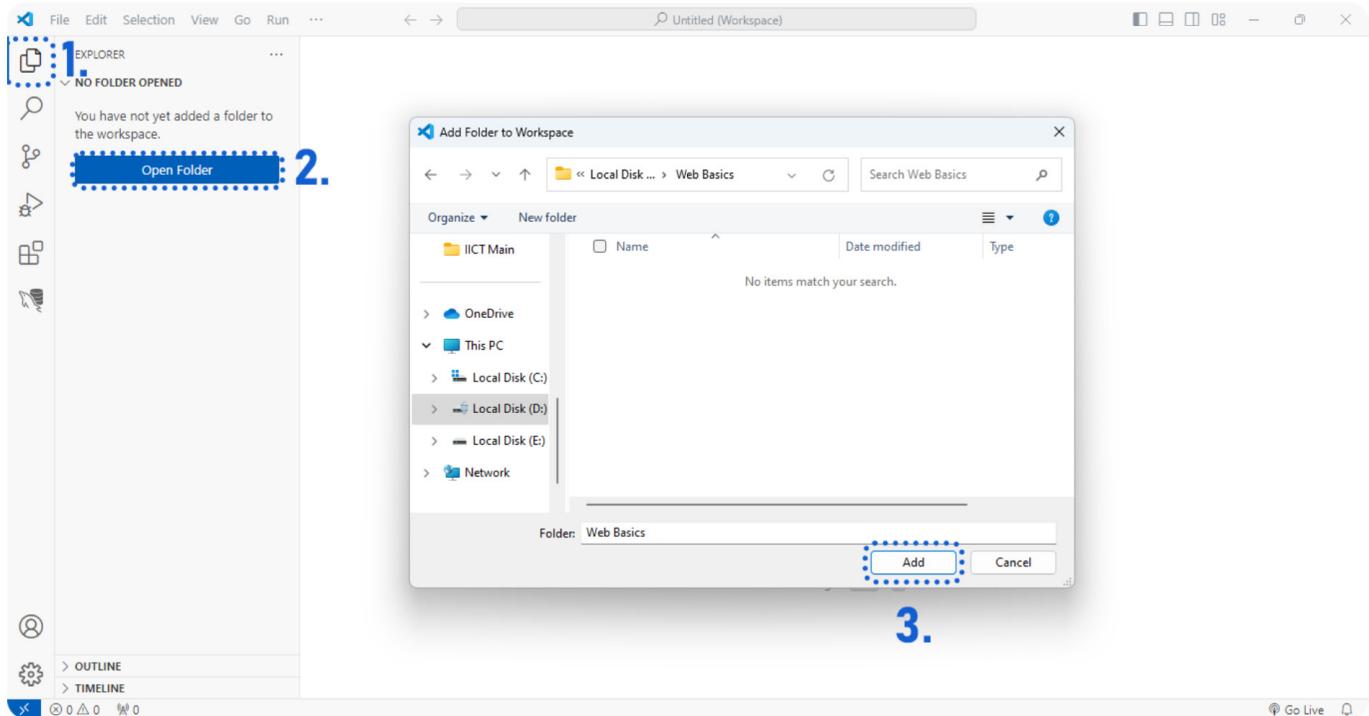
2.2 Installing Live Server Extension for VS Code

Follow the same steps as before to install the **Live Server** extension. Search **Live Server** in the Marketplace.

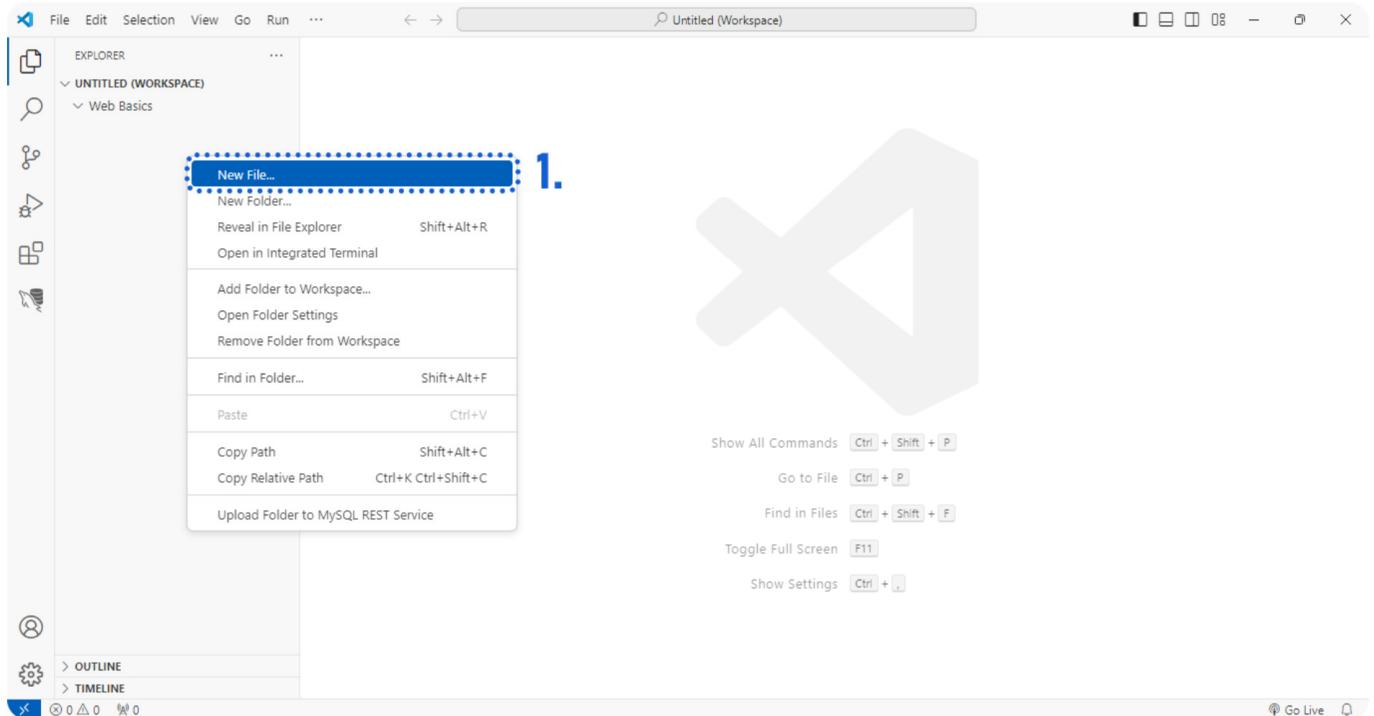
The screenshot shows the Visual Studio Code Marketplace interface again. The sidebar on the left shows the 'live server' search results, with the first result, 'Live Server' by Ritwick Dey, highlighted. The main area shows the details for the 'Live Server' extension, version v5.7.9. It features a purple icon of two overlapping circles, the extension name, its rating (4.5 stars), and download count (54,231,790). Below this, there are tabs for 'DETAILS', 'FEATURES', and 'CHANGELOG'. A sidebar on the right lists categories like 'Other' under 'Categories' and 'Marketplace', 'Issues', 'Repository', 'License', and 'Ritwick Dey' under 'Resources'. At the bottom, there's a 'More Info' section with details such as publication date (2017-06-28), last release date (2022-08-31), and identifier (ritwickdey.liveserver).



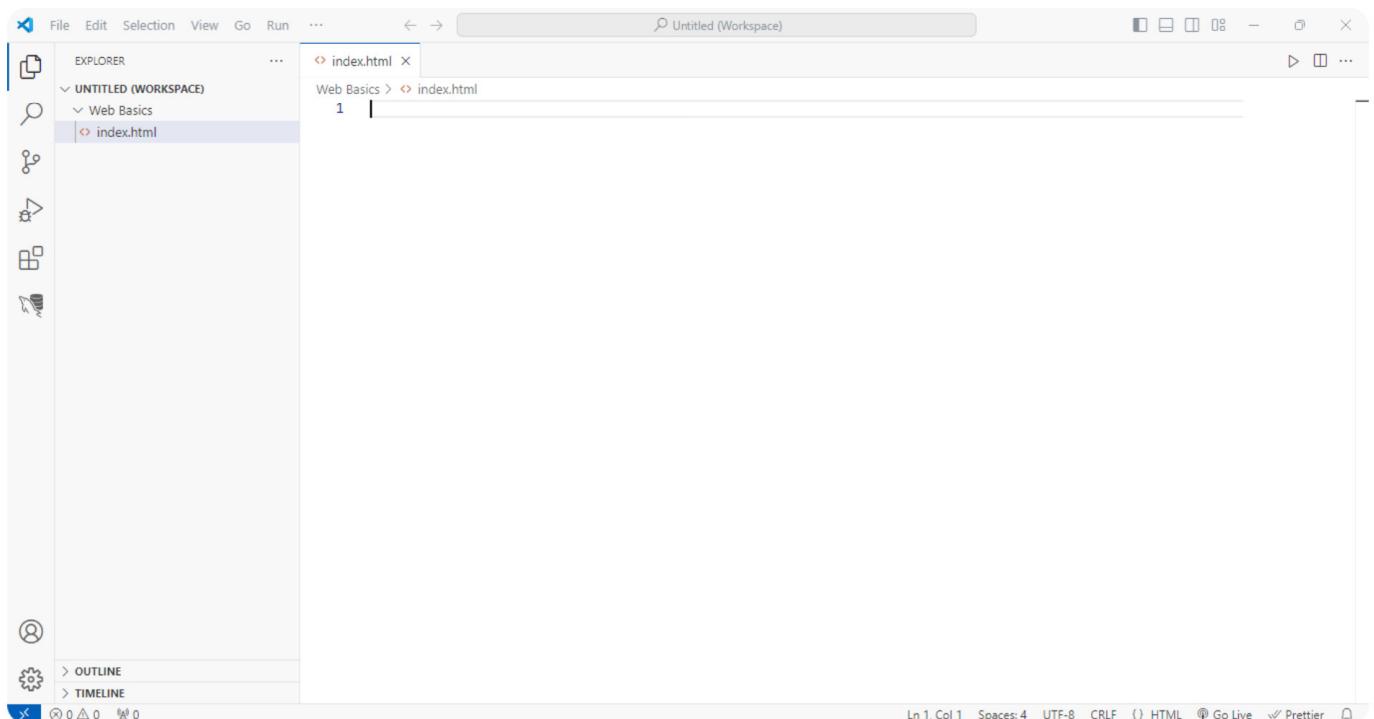
5. Click on **Explorer** icon. Or you can press **Ctrl + Shift + E** to open the explorer.
6. Select the folder in your desire location to create the environment.
7. Click on **Add**.



8. Under the **Explorer** pane, and in the folder, you have created, **right-click** and add a new file by clicking on **New File...**



9. Type the name for your HTML file without spaces and give it the **.html** extension.
10. Press **Enter**



2.3 Prettier - Code Formatter

Prettier is a **code formatter** that helps maintain a **consistent style** in your code. It **automatically formats** your **JavaScript, HTML, CSS**, and other languages based on predefined or custom style rules. Some key features of Prettier include:

- Automatic formatting on file saves, making sure your code follows best practices and looks clean.
- Supports a wide variety of languages and is highly configurable.
- Enforces a uniform code style, reducing time spent on code reviews related to formatting issues.

Prettier is great for ensuring that the code across a project looks the same, regardless of how different developers write it.



2.4 Live Server

The **Live Server** extension creates a **local development server** with **live reloading** for **static** and **dynamic web pages**. Some of its features include:

- It automatically refreshes the browser whenever you save changes to HTML, CSS, or JavaScript files.
- Allows you to see immediate updates and feedback on your work, enhancing productivity and making debugging faster.
- It's particularly useful in front-end development, offering a preview of how your site will appear in a real browser environment.



Both Prettier and Live Server are commonly used together to streamline the development process in VS Code, offering code formatting and live preview capabilities.

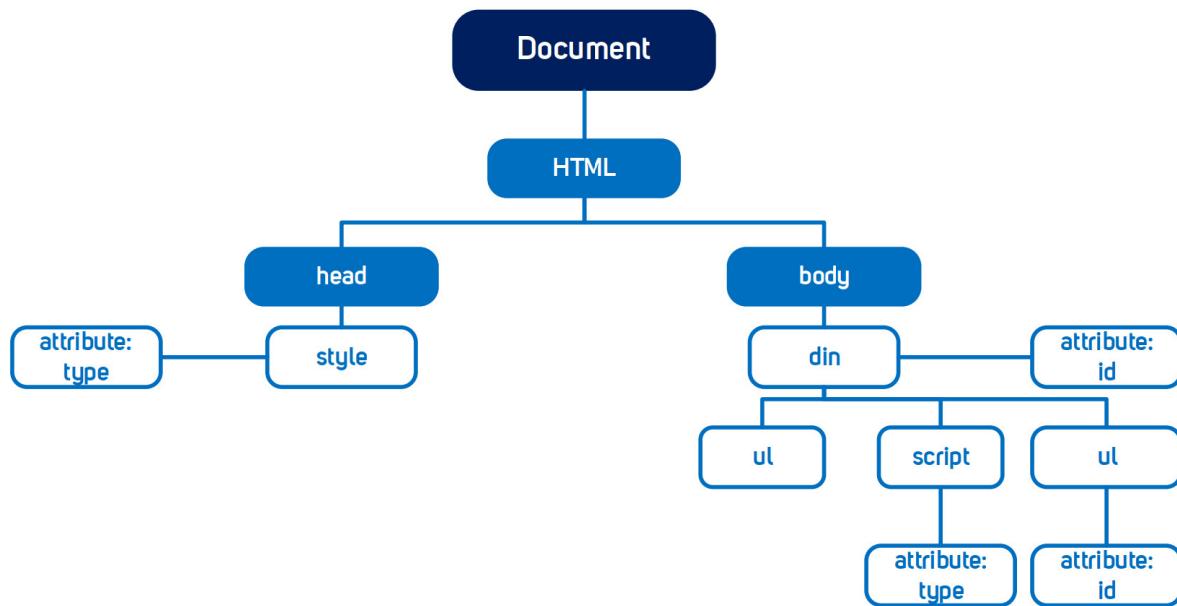
4. Document Object Model (DOM)

The **Document Object Model (DOM)** is a **cross-platform** and **language-independent** interface that represents a **web page's structure, content, and styling** as a **tree-like model**. Each part of the document—whether it's an HTML element, an attribute, or a piece of text—is represented as an **object (or "node")** in this tree.

The DOM allows developers to programmatically access and manipulate the **structure, style, and content** of a **web document**. **Browsers automatically** create a **DOM** from the **HTML code** when a **page loads**. Through the DOM, a web document can be **updated dynamically without reloading the entire page**.

This representation is what makes it possible to modify elements, attributes, or content on a page using client-side languages like JavaScript, but the DOM itself is not tied to any specific language.

In essence, the DOM provides a structured way to model documents and enables interaction between code (such as JavaScript) and the visual representation of a webpage.



5. HTML & CSS in Action

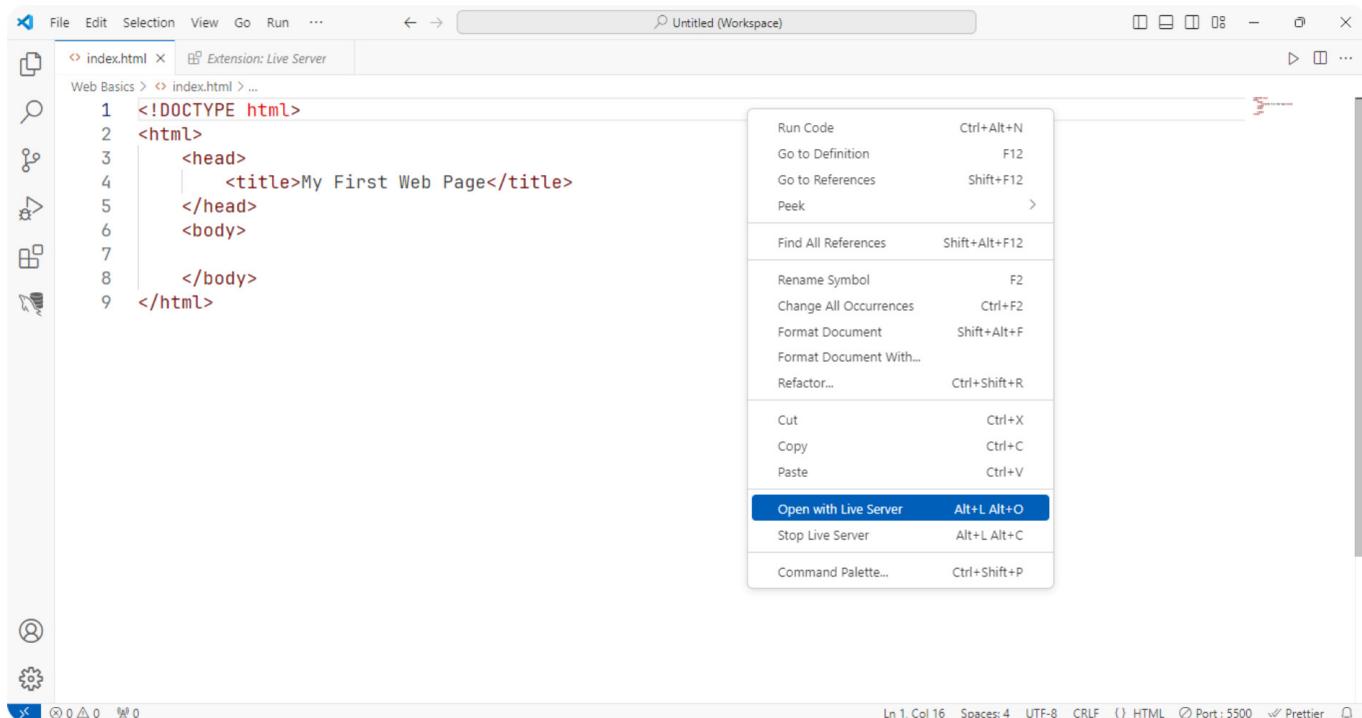
```
<!DOCTYPE html>
<html>
  <head>
    <title>My First Web Page</title>
  </head>
  <body>

    </body>
</html>
```

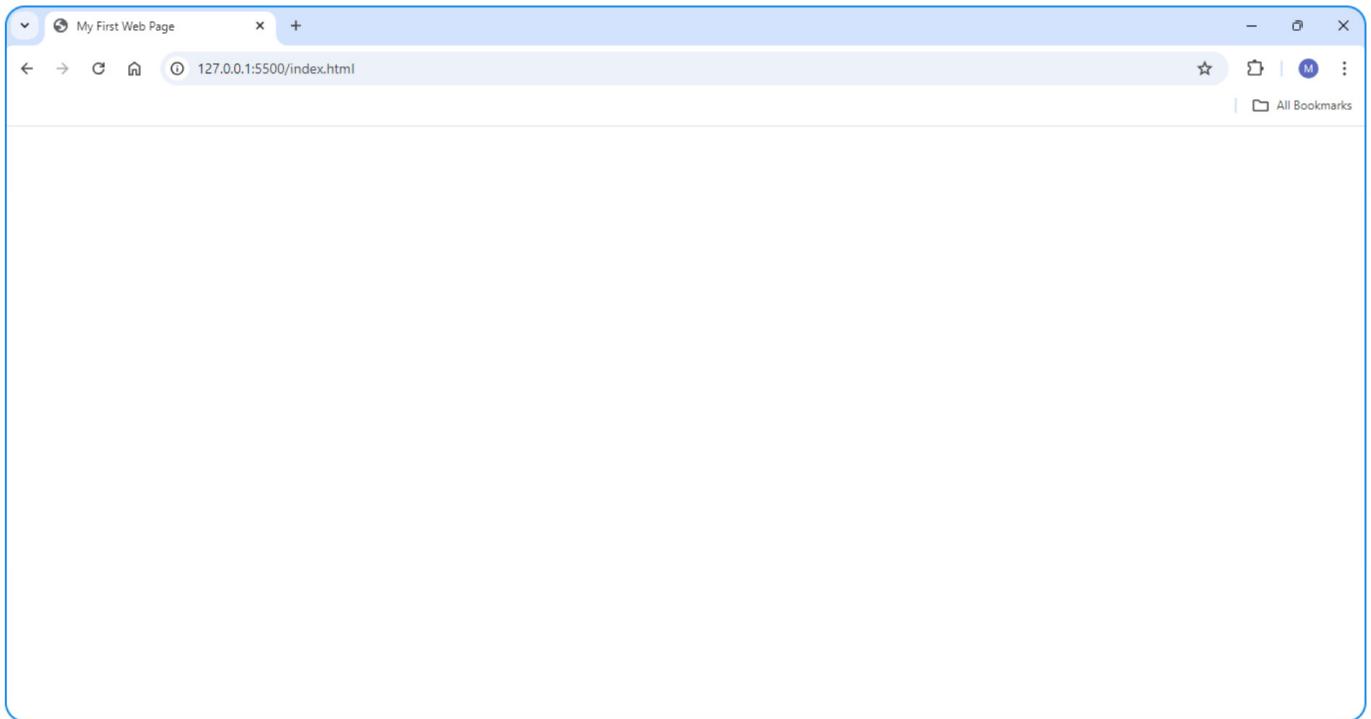
1. **<!DOCTYPE html>**: This declaration defines the document type and version of HTML being used. In this case, it indicates that the document is written in HTML5. It's not technically an HTML tag but ensures the browser renders the page in standards-compliant mode.
2. **<html>**: This is the root element of the HTML document. All other HTML elements are contained within this tag. It defines the beginning and end of the HTML content.
3. **<head>**: The head section contains metadata about the document, such as the title, links to CSS files, and other information that doesn't appear directly on the web page. It is primarily used for behind-the-scenes content.

4. **<title>My First Web Page</title>**: This element defines the title of the web page. It is displayed on the browser's title bar or tab. In this example, the title is "My First Web Page."
5. **<body>**: The body tag contains the content of the HTML document that is visible to the user, such as text, images, links, and other elements. Currently, it is empty in this example, but content can be placed between the **<body>** tags to display on the web page.
6. **</html>**: This is the closing tag of the root **<html>** element, signalling the end of the HTML document.

To see the page, right-click anywhere in the VS Code editor section and click on “Open with Live Server”.

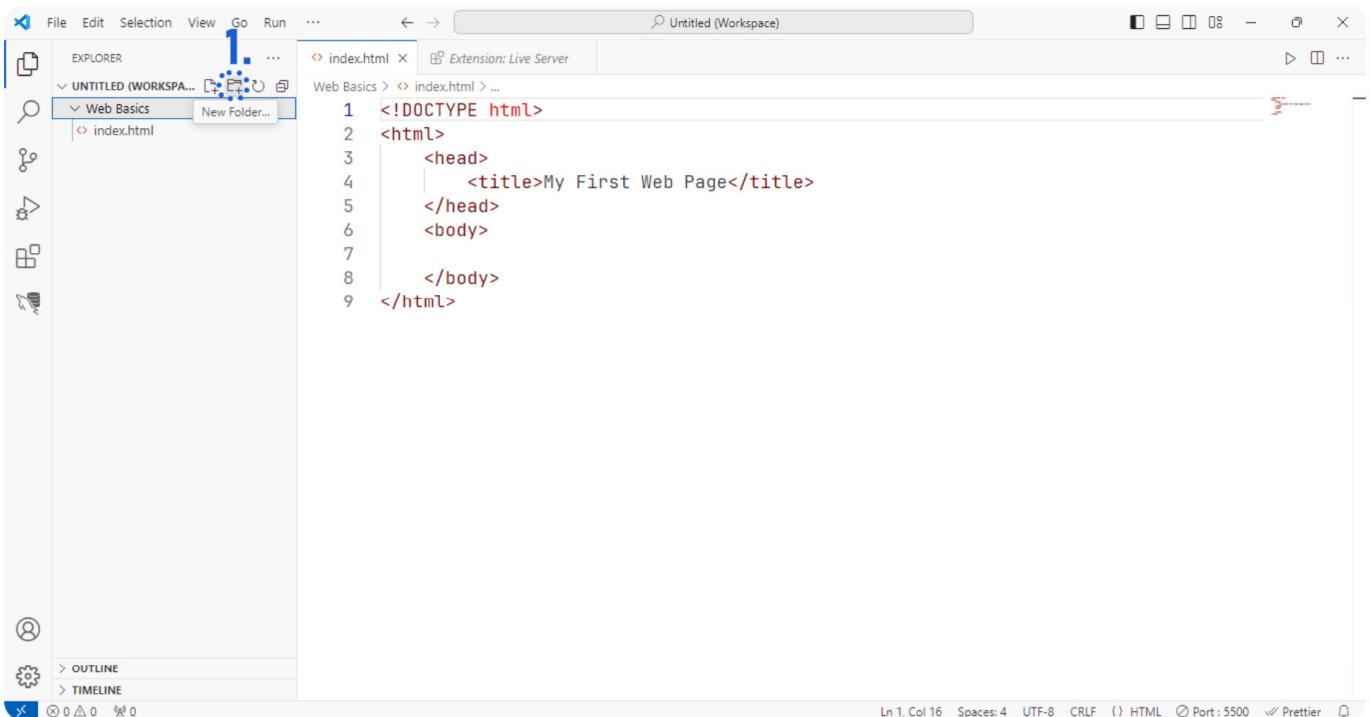


The web page will open in your default browser.



5.1 Image and Text Element

1. Create a new folder inside the current working folder (i.e., **Web Basics** in this case)
2. Rename the folder (i.e., **images** in this case)
3. Drag and drop an **image** to add it to your web page in the newly created folder.



```
<!DOCTYPE html>
<html>
  <head>
    <title>My First Web Page</title>
  </head>
  <body>
  </body>
</html>
```

File Edit Selection View Go Run ...

UNTITLED (WORKSPACE) ...

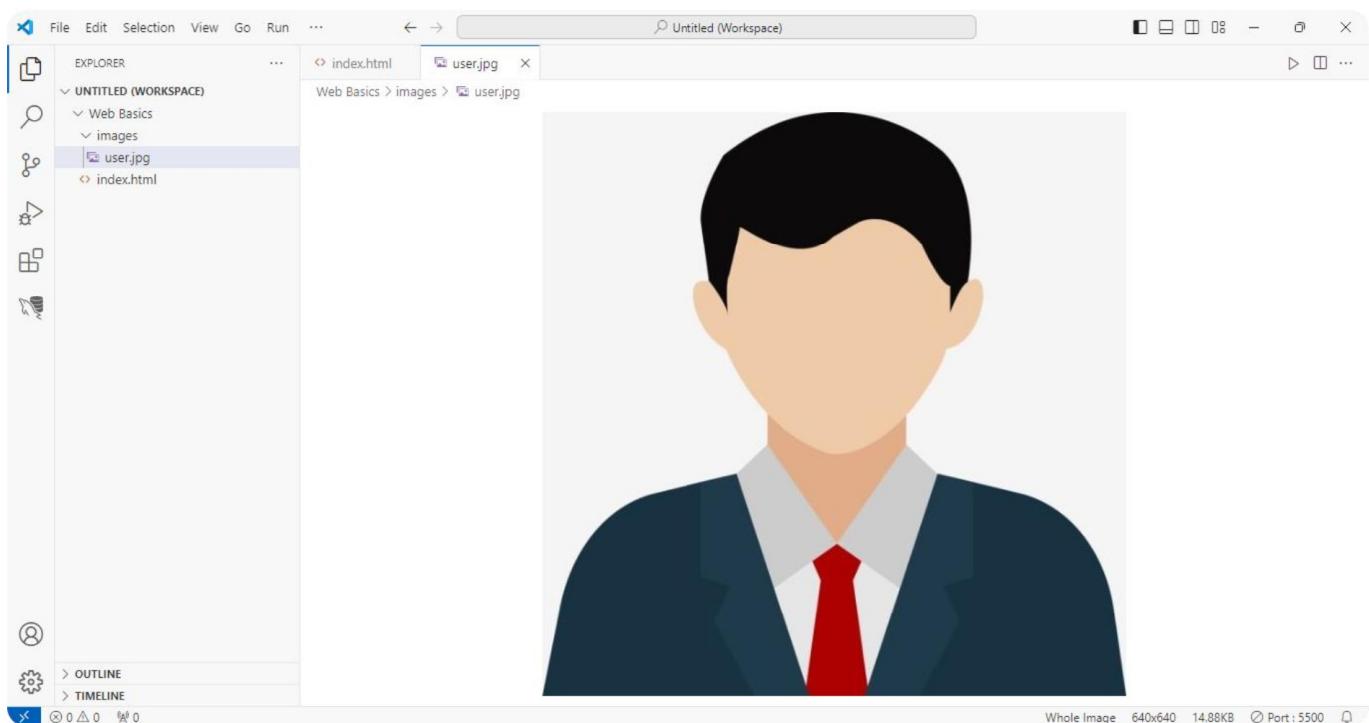
index.html Extension: Live Server

Web Basics > index.html > ...

2. 1 <!DOCTYPE html>

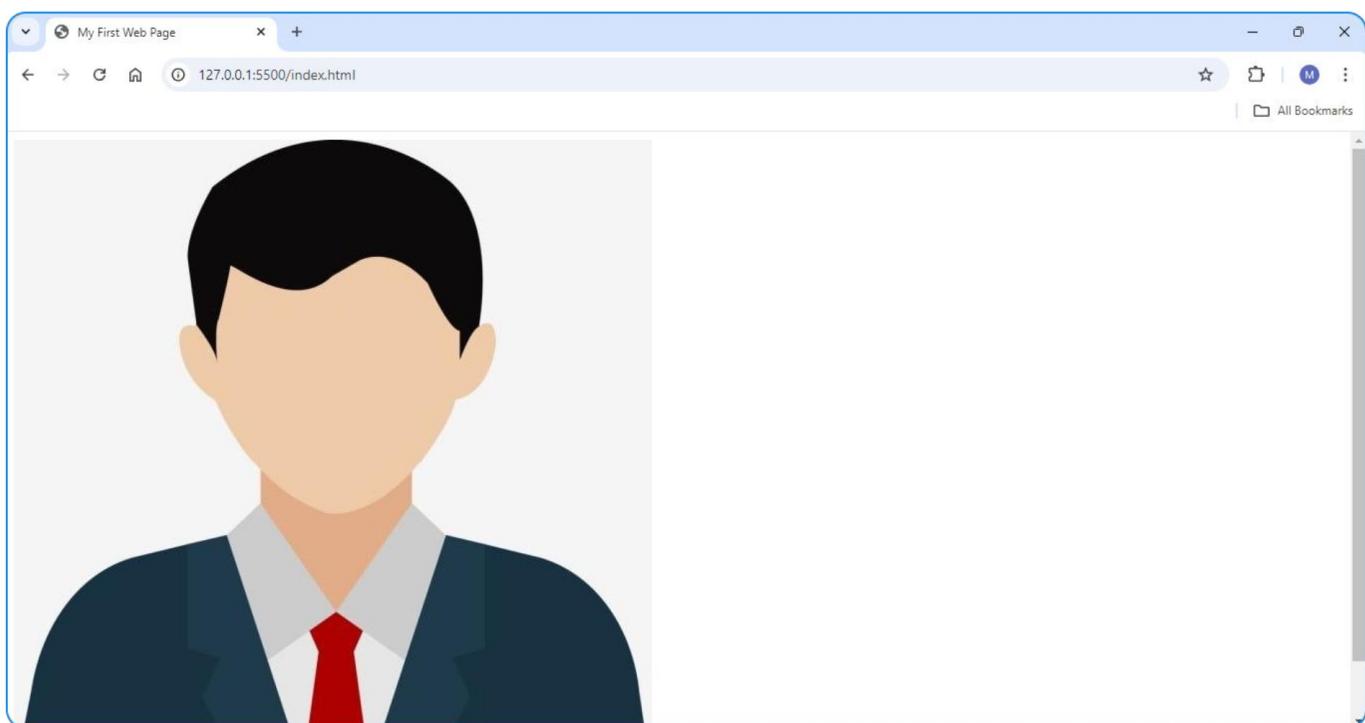
```
<html>
  <head>
    <title>My First Web Page</title>
  </head>
  <body>
    </body>
</html>
```

Ln 1, Col 16 Spaces: 4 UTF-8 CRLF ⓘ HTML ⚡ Port: 5500 ✅ Prettier



5.1.1 Image Element

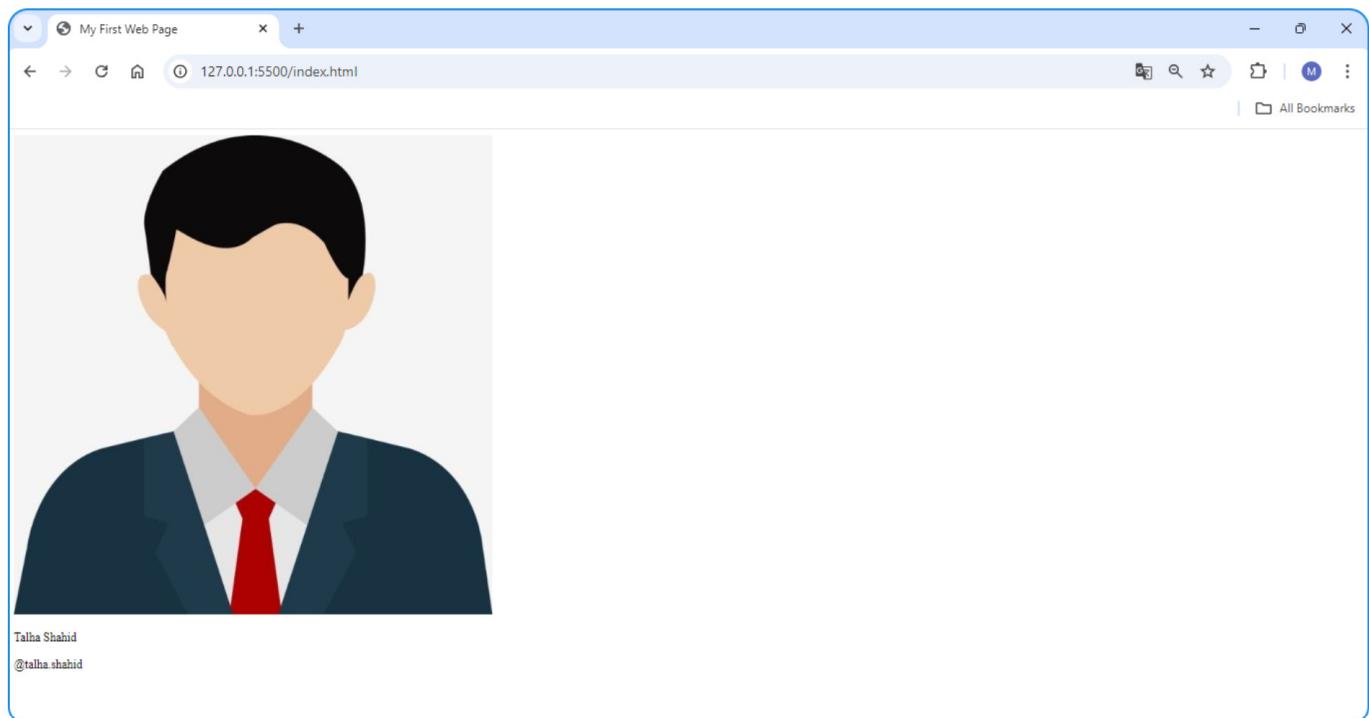
```
<!DOCTYPE html>
<html>
  <head>
    <title>My First Web Page</title>
  </head>
  <body>
    
  </body>
</html>
```



- ****: This is an HTML element used to embed an image in a webpage.
- **src="images/user.jpg"**: The **src** (source) attribute specifies the path to the image file. In this case, it's looking for an image named **user.jpg** in the **images** folder
- **alt="An image of the user"**: The **alt** attribute provides alternative text for the image. If the image cannot be displayed (e.g., if the file is missing), this text will be shown instead. It's also useful for accessibility, allowing screen readers to describe the image to visually impaired users.

5.1.2 Text Element

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First Web Page</title>
  </head>
  <body>
    
    <p>Talha Shahid</p>
    <p>@talha.shahid</p>
  </body>
</html>
```



- **<p>**: This HTML tag is used to define a paragraph of text. It creates a block of content that will have space (or margin) before and after it.
- In the first **<p>** element, the text "**Talha Shahid**" is displayed as a separate paragraph.
- In the second **<p>** element, the text "**@talha.shahid**" is displayed in another paragraph.

How the **<p>** Tag Works:

Paragraphs in HTML automatically get some spacing around them, which helps in laying out content neatly. Even though the paragraphs are very short in this example, they still benefit from this structure, making the text easier to read and well-organized on the webpage.

In this case, the two **<p>** elements create two lines of text:

"Talha Shahid"

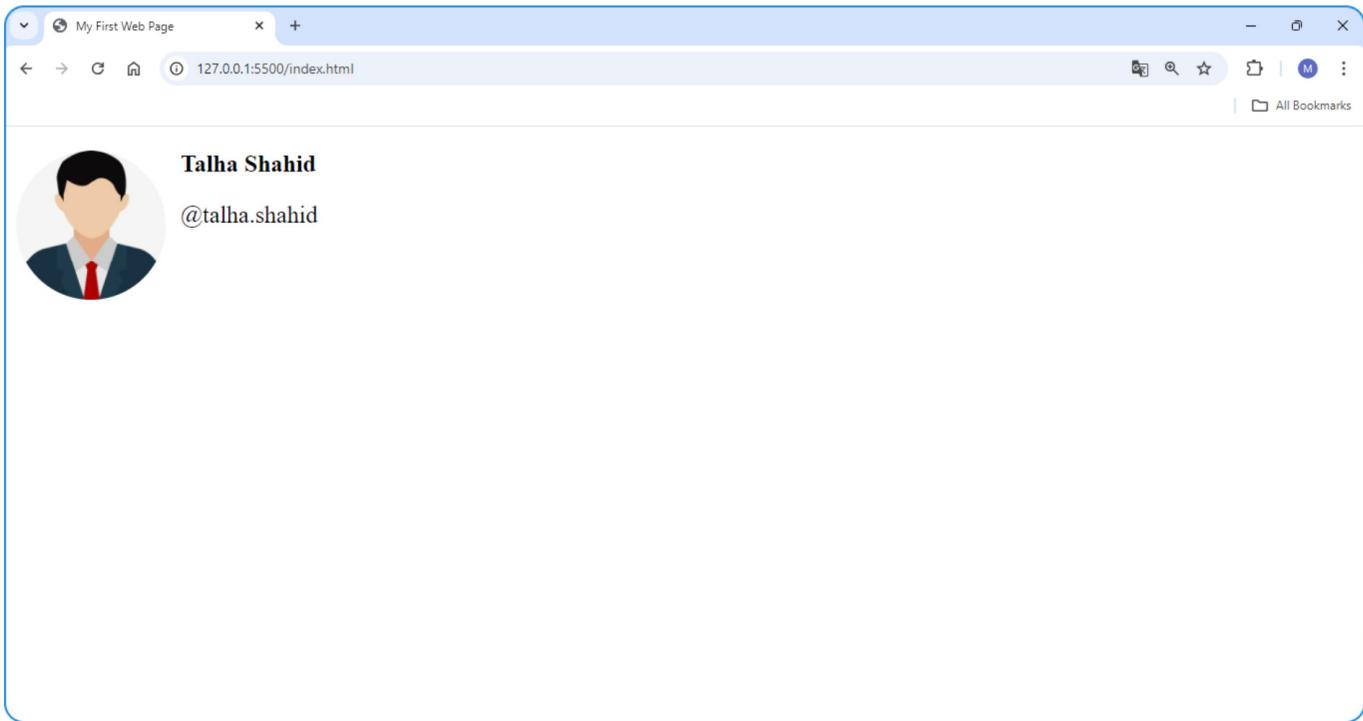
"@talha.shahid"

Each is on its own line, separated as distinct paragraphs.

5.1.3 Add Styling | Here comes CSS

As it can be seen that, the image and texts in our web page is looking odd. Let's apply CSS to style them properly.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First Web Page</title>
  </head>
  <style>
    img {
      width: 100px;
      border-radius: 50px;
      float: left;
      margin-right: 10px;
    }
    p.username {
      font-weight: bold;
    }
  </style>
  <body>
    
    <p class="username">Talha Shahid</p>
    <p>@talha.shahid</p>
  </body>
</html>
```



img Selector:

width: 100px;:

This sets the width of all `` elements to **100** pixels, ensuring a uniform size for images on the page.

border-radius: 50px;:

This property rounds the corners of the image, giving it a circular appearance. This is particularly effective for profile images, making them visually appealing.

float: left;:

This allows the image to float to the left side of the container, letting other elements (like text) wrap around it. This creates a more integrated layout, especially useful for user profiles.

margin-right: 10px;:

Adds a **10**-pixel space to the right of the image, ensuring that there is breathing room between the image and any adjacent text, which enhances readability.

p.username Selector:

font-weight: bold;:

This rule applies to any paragraph (`<p>`) element with the class **username**. The use of bold styling emphasizes the text within that paragraph, making it stand out from other text on the page. By adding `class="username"` to the paragraph tag, you ensure that the name "**Talha Shahid**" appears prominently, drawing the viewer's attention to it. This is particularly useful in a user profile context, where you want the user's name to be easily identifiable.

6. Formatting Code

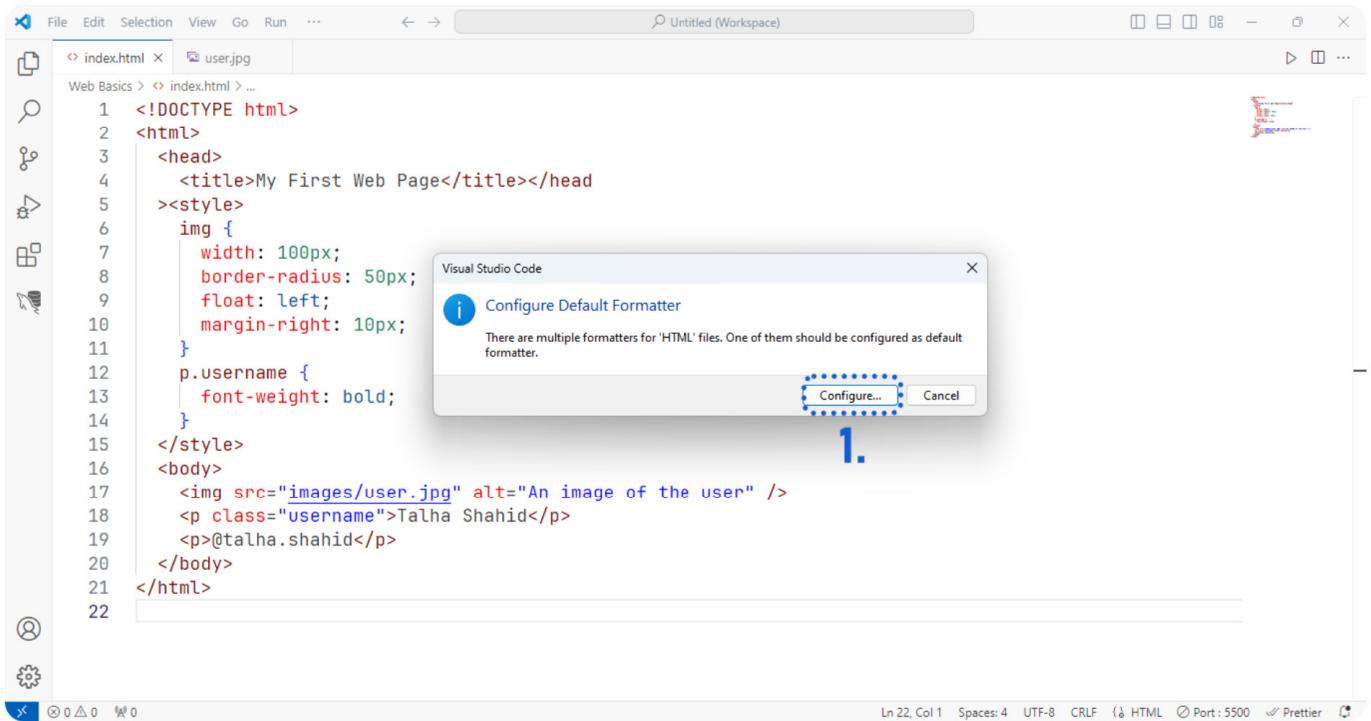
1. Right-click inside the .html file.
2. Click on Command Palette...

A screenshot of the Visual Studio Code interface. The left sidebar shows icons for file operations like Open, Save, Find, and Settings. The main editor area contains an HTML file named 'index.html'. The code includes an image element with a width of 100px, border-radius of 50px, and float set to left. A CSS class 'username' is defined with a bold font-weight. The command palette is open on the right, with 'Format Document' highlighted. Other options in the palette include Run Code, Go to Definition, Go to References, Find All References, Rename Symbol, Change All Occurrences, Format Document, Refactor..., Cut, Copy, Paste, Open with Live Server, Stop Live Server, and Command Palette... (which is also highlighted).

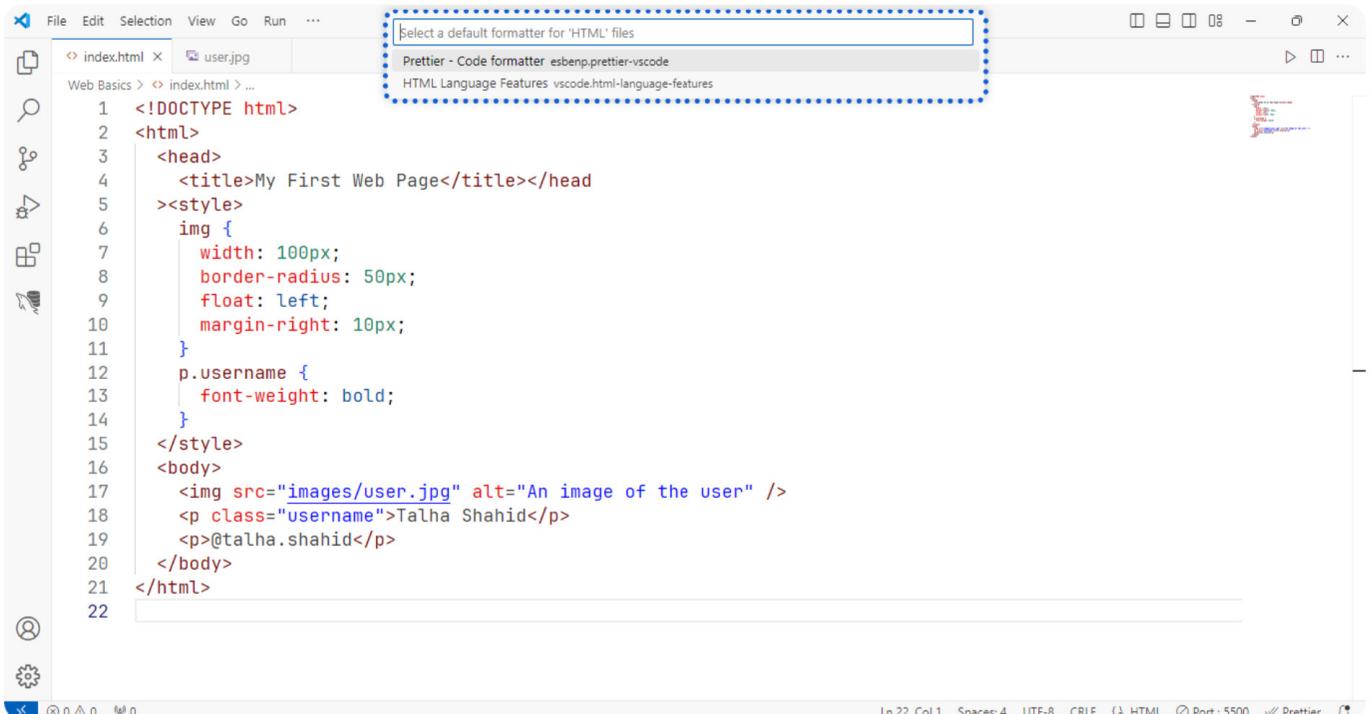
3. Type **Format Document** and click on it

A screenshot of the Visual Studio Code interface, similar to the previous one but with the code in 'index.html' expanded. The command palette dropdown is open at the top, showing 'Format Document' as the selected item. Other options like 'Format Document (Forced)' and 'Format Document With...' are visible. The status bar at the bottom indicates the code is at line 22, column 1.

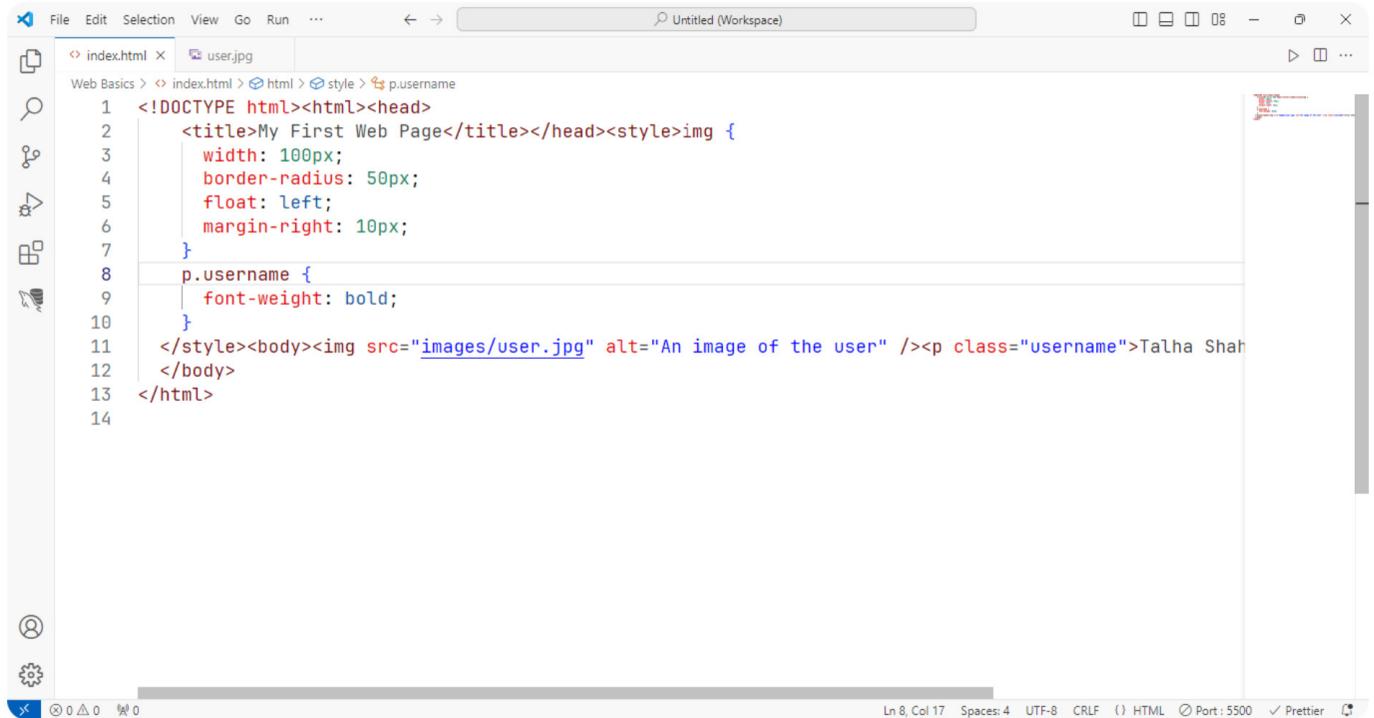
4. A dialogue window will open, click on **Configure...**



5. Select Prettier – Code formatter.



6. As you can see the code is not formatted properly in the following image.



A screenshot of a code editor window titled "Untitled (Workspace)". The left sidebar shows two files: "index.html" and "user.jpg". The main editor area contains the following HTML and CSS code:

```
<!DOCTYPE html><html><head>
<title>My First Web Page</title></head><style>img {
    width: 100px;
    border-radius: 50px;
    float: left;
    margin-right: 10px;
}
p.username {
    font-weight: bold;
}
</style><body><p class="username">Talha Shah
</body>
</html>
```

The code is displayed in a monospaced font, indicating it is not yet formatted.

7. To do so, click on the **gear** icon from the bottom left, and click on **Settings...**

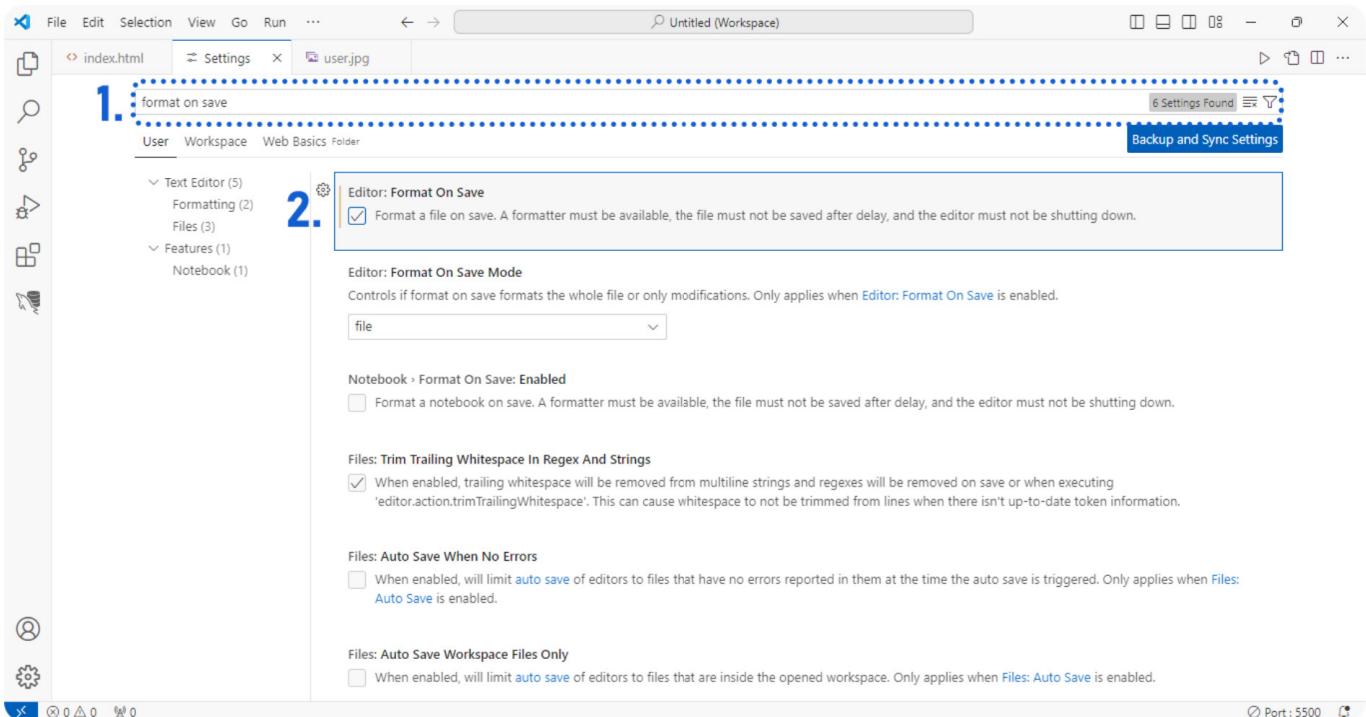


A screenshot of a code editor window titled "Untitled (Workspace)". The left sidebar shows two files: "index.html" and "user.jpg". The main editor area contains the same HTML and CSS code as the previous screenshot. A gear icon in the bottom-left corner has opened a dropdown menu. The "Settings" option in this menu is highlighted with a blue background and white text. The number "2." is overlaid on the left side of the menu.

2.

- Command Palette... Ctrl+Shift+P
- Profiles
- Settings** Ctrl+,
- Extensions Ctrl+Shift+X
- Keyboard Shortcuts Ctrl+K Ctrl+S
- Snippets
- Tasks
- Themes >
- Backup and Sync Settings...
- Check for Updates...

8. Type format document in search bar and click on Format On Save.



9. Close the Settings Tab.

10. Press **ctrl + s and your code will be formatted automatically.**

The screenshot shows the VS Code editor with an open file named 'index.html'. The code is a simple HTML page with a title, a style block containing CSS for an image and a paragraph, and a body block containing an image and two paragraphs. The code is well-formatted with proper indentation and syntax highlighting. The status bar at the bottom indicates the file is in 'HTML' mode, has port 5500 open, and Prettier is active.

```
<!DOCTYPE html>
<html>
<head>
<title>My First Web Page</title>
</head>
<style>
img {
    width: 100px;
    border-radius: 50px;
    float: left;
    margin-right: 10px;
}
.p.username {
    font-weight: bold;
}
</style>
<body>

<p class="username">Talha Shahid</p>
<p>@talha.shahid</p>
</body>
</html>
```

7. Basic HTML Boilerplate

Type the **exclamation** character and press tab, the following code will be inserted that you will need as a starting code for your HTML document.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

1. **<!DOCTYPE html>:**

- This declaration defines the document type and version of HTML being used. It tells the browser to expect an HTML5 document. This is essential for ensuring that the page is rendered correctly.

2. **<html lang="en">:**

- This tag encloses all the HTML content and specifies the language of the document. The **lang="en"** attribute indicates that the primary language is English, which can help search engines and assistive technologies understand the content better.

3. **<head> Section:**

- This section contains meta-information about the document:
- **<meta charset="UTF-8">:**
 - This tag sets the character encoding for the document to **UTF-8**, which is the most widely used encoding on the web. It supports all characters and symbols from various languages, ensuring proper display of text.
- **<meta name="viewport" content="width=device-width, initial-scale=1.0">:**
 - This tag is crucial for responsive web design. It tells the browser how to control the page's dimensions and scaling. Setting **width=device-width** ensures that the page is rendered at the width of the device's screen, and **initial-scale=1.0** sets the initial zoom level when the page is first

loaded. This is particularly important for mobile devices, ensuring a good user experience.

- **<title>Document</title>:**

- This tag specifies the title of the web page, which appears in the browser tab and is used by search engines. While "Document" is a placeholder, it should be replaced with a meaningful title that reflects the content of the page.

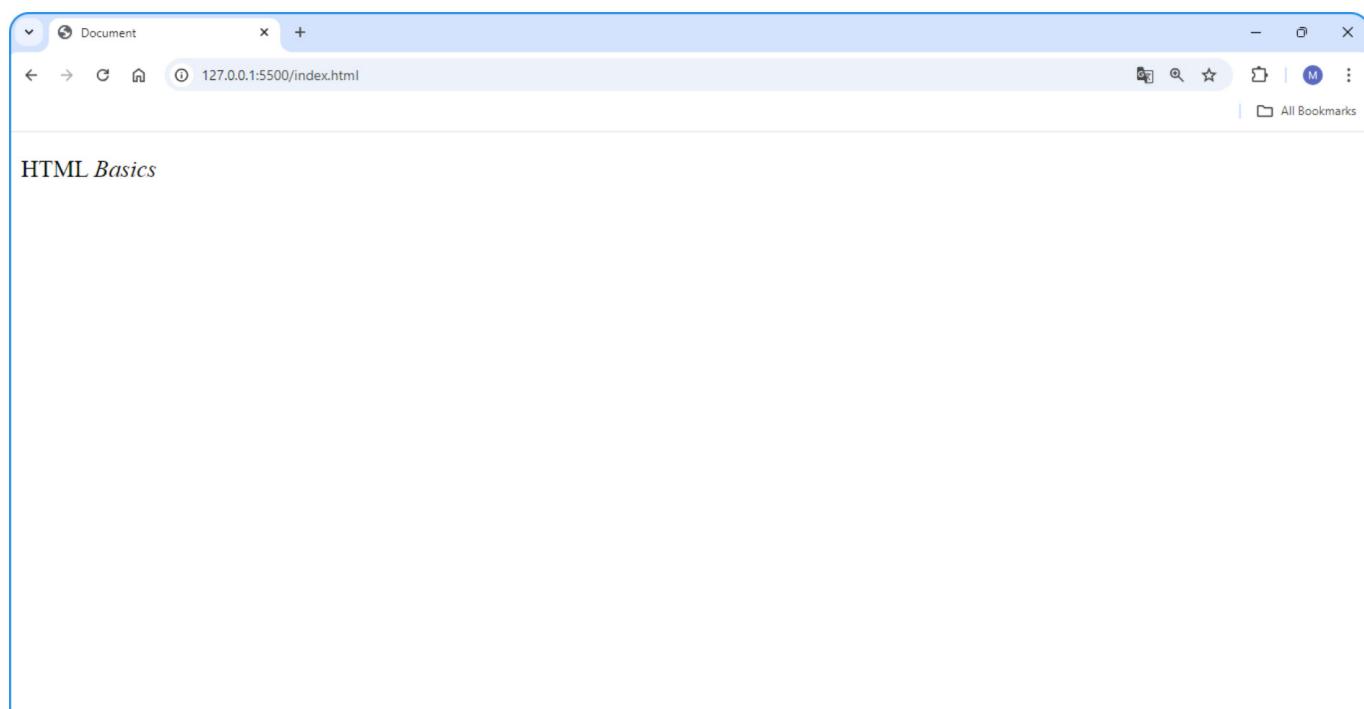
4. <body> Section:

- This section is where the content of the web page will go. Currently, it is empty, but it is the main area where text, images, links, and other elements will be added to create the user interface.

8. Working with Text

8.1 Emphasis Tag

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <p>HTML <em>Basics</em></p>
```

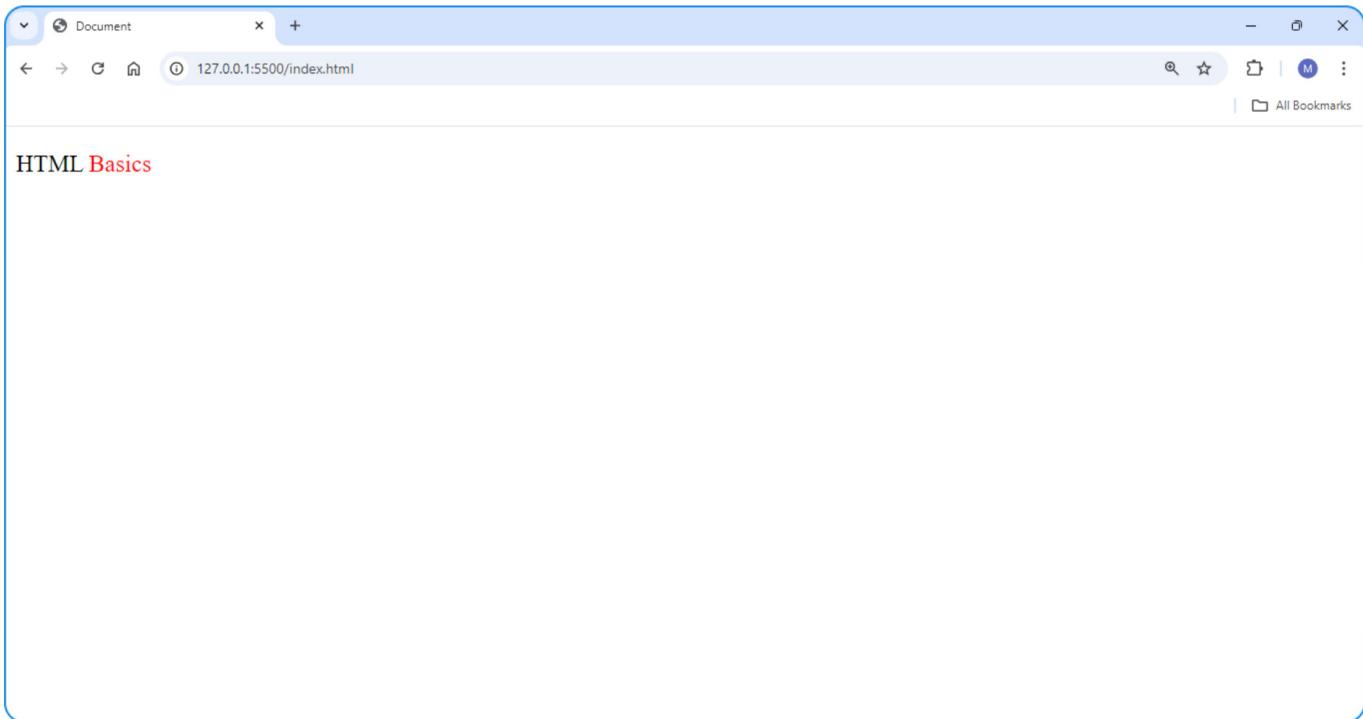


```
</body>
</html>
```

- The `` tag is used to emphasize a portion of text—in this case, the word "**Basics.**" Here's how it works:
- The `` tag is semantically used to indicate that the text within it should be emphasized. This typically implies that the content has increased importance or is meant to stand out in some way.
- By default, browsers usually render text within the `` tag in italics, which visually differentiates it from surrounding text. However, this styling can be overridden with CSS if needed.
- The emphasis provided by the `` tag is not just visual. Screen readers and other assistive technologies recognize the tag and often provide vocal emphasis when reading the text aloud, making it more accessible for users with visual impairments.
- Using semantic HTML elements like `` improves the overall structure and meaning of web content. It helps search engines and user agents understand the context and significance of different parts of a webpage.

8.2 Styling the Emphasis Tag

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
  <title>Document</title>
  <style>
    em {
      color: red;
      font-style: normal;
    }
  </style>
</head>
<body>
  <p>HTML <em>Basics</em></p>
</body>
</html>
```



9. Headings

In HTML, headings are defined using six levels of heading tags, from **<h1>** to **<h6>**, with each level indicating a different degree of importance. Here's a brief overview of each:

1. **<h1>**: Represents the most important heading on the page, typically used for the main title. There should be only one **<h1>** per page for SEO purposes.
2. **<h2>**: Used for subheadings that fall under the main heading. It helps to break up content into sections and indicate a hierarchy.
3. **<h3>**: A subheading of **<h2>**, used for further subdivisions within a section. It continues to structure content hierarchically.
4. **<h4>**: A subheading of **<h3>**, used for even more specific sections. It allows for detailed breakdowns within content.
5. **<h5>**: A subheading of **<h4>**, generally used for lower-level headings. It's useful for minor points or topics.
6. **<h6>**: Represents the least important heading. It's typically used for the most detailed subdivisions within the content hierarchy.

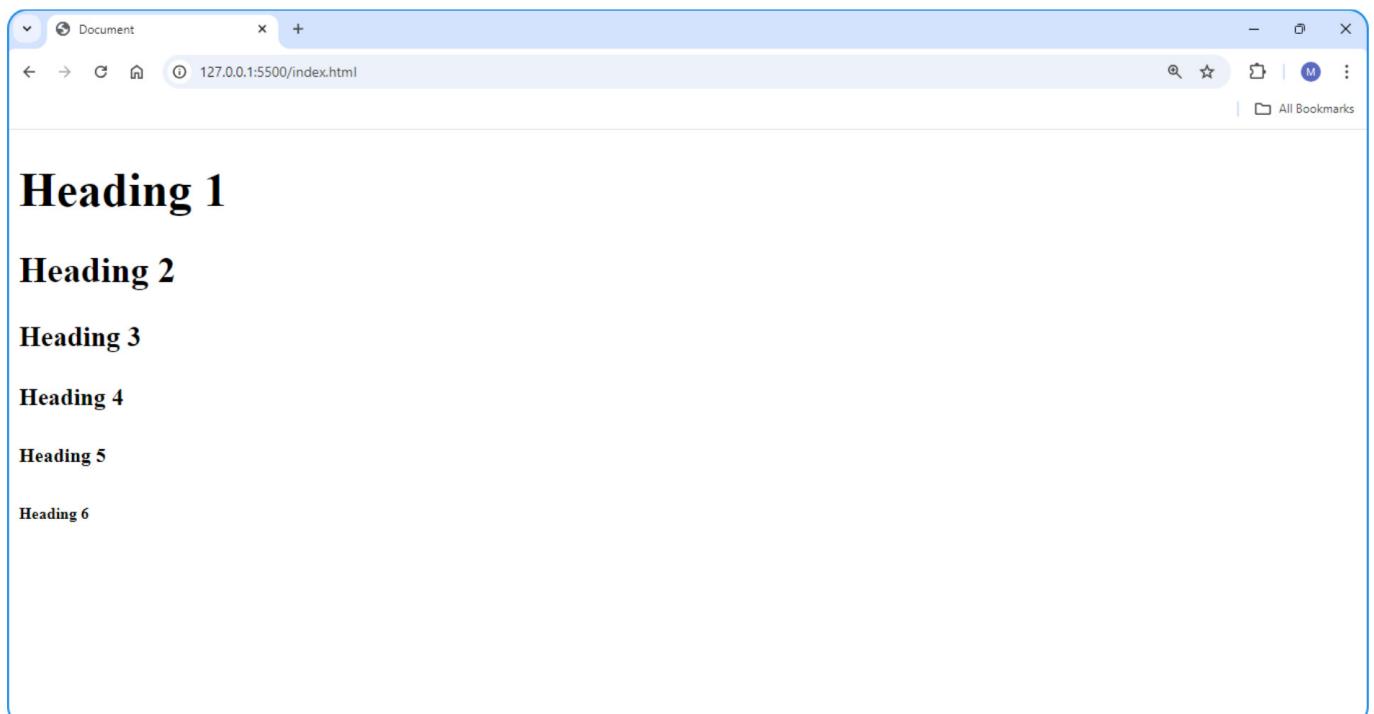
9.1 Importance of Headings

Accessibility: Headings help screen readers navigate the page more easily, improving accessibility.

SEO: Properly structured headings can enhance search engine optimization by indicating the structure and importance of content.

Readability: They improve the overall readability of the content, making it easier for users to scan and find information quickly.

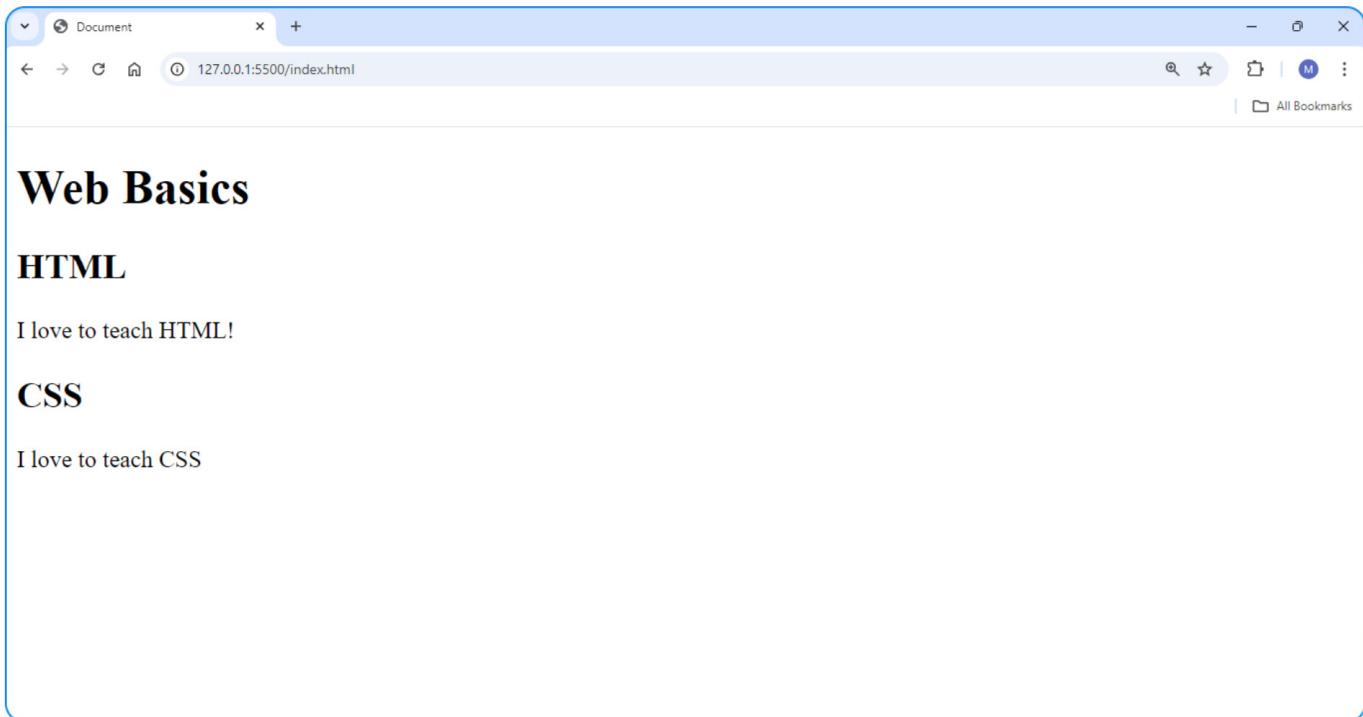
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
    <h3>Heading 3</h3>
    <h4>Heading 4</h4>
    <h5>Heading 5</h5>
    <h6>Heading 6</h6>
  </body>
</html>
```



9.1.1 Key Points Regarding the Heading Tags

- Hierarchy:** Heading tags (**<h1>** to **<h6>**) are organized based on their hierarchy, not just size. **<h1>** is the most important, while **<h6>** is the least.
- Usage:** **<h1>** should be used only once per page to define the main topic. Subsequent headings, like **<h2>** and **<h3>**, are for subtopics and details.
- Logical Structure:** Avoid using **<h2>** if there's no **<h1>**. This maintains a logical flow and helps with content accessibility and organization.
- SEO Benefits:** Proper heading hierarchy improves search engine optimization by clearly defining the content structure for search engines.
- Accessibility:** Correct use of heading tags enhances accessibility for users who rely on screen readers, allowing them to navigate the content effectively.

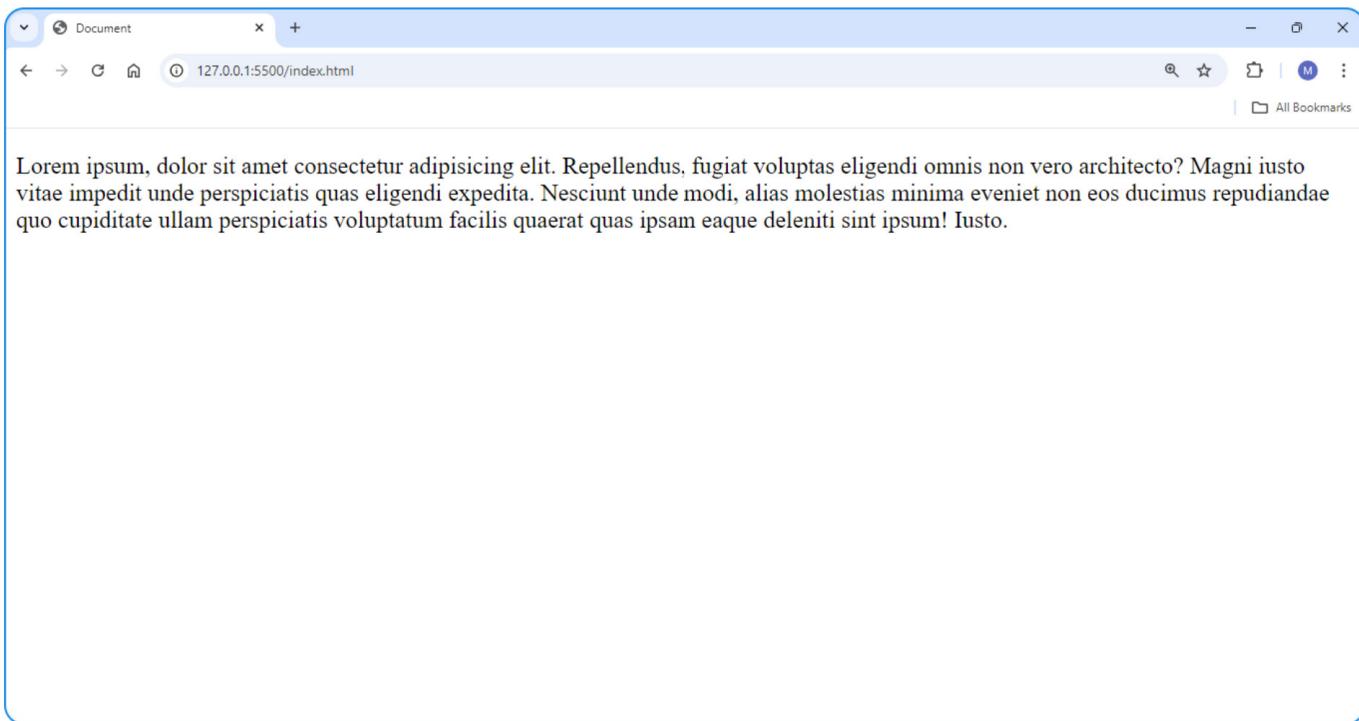
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <h1>Web Basics</h1>
    <h2>HTML</h2>
    <p>I love to teach HTML!</p>
    <h2>CSS</h2>
    <p>I love to teach CSS</p>
  </body>
</html>
```



10. None-Breaking Space in Paragraphs (HTML Entities)

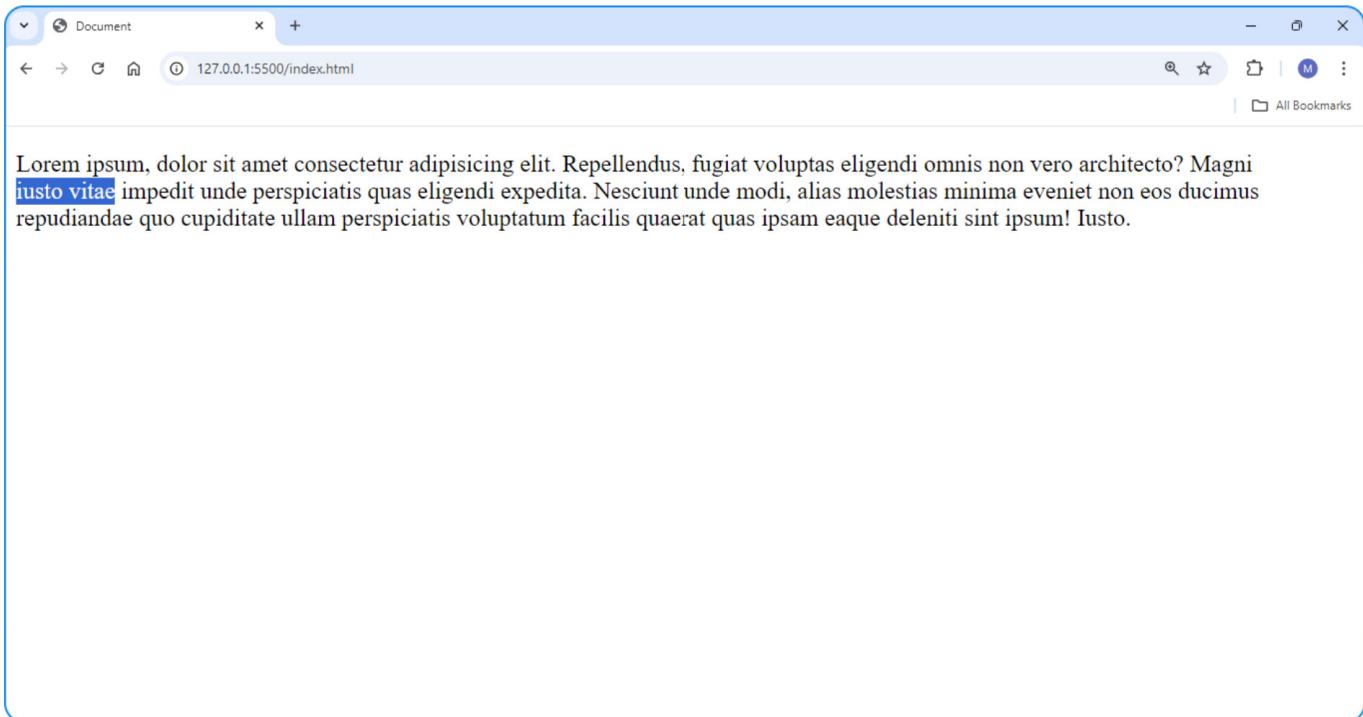
First, insert a dummy paragraph by typing **lorem50** inside the paragraph tag. This will generate a dummy paragraph of **50** words.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <p>
      Lorem ipsum, dolor sit amet consectetur adipisicing elit.
      Repellendus, fugiat voluptas eligendi omnis non vero
      architecto? Magni iusto vitae impedit unde perspiciatis
      quas eligendi expedita. Nesciunt unde modi, alias molestias
      minima eveniet non eos ducimus repudiandae quo cupiditate
      ullam perspiciatis voluptatum facilis quaerat quas ipsam
      eaque deleniti sint ipsum! Iusto.
    </p>
  </body>
</html>
```



If we want that the word “**iusto**” at last in the first line and the word “**vitae**” at first in the second line will always come together in the paragraph, we will type (none-breaking space) between these two words.

```
</html><!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <p>
      Lorem ipsum, dolor sit amet consectetur adipisicing elit.
      Repellendus, fugiat voluptas eligendi omnis non vero
      architecto? Magni iusto&nbsp;vitae impedit unde
      perspiciatis quas eligendi expedita. Nesciunt unde modi,
      alias molestias      minima eveniet non eos ducimus
      repudiandae quo cupiditate ullam perspiciatis voluptatum
      facilis quaerat quas ipsam eaque deleniti sint ipsum!
      Iusto.
    </p>
  </body>
</html>
```



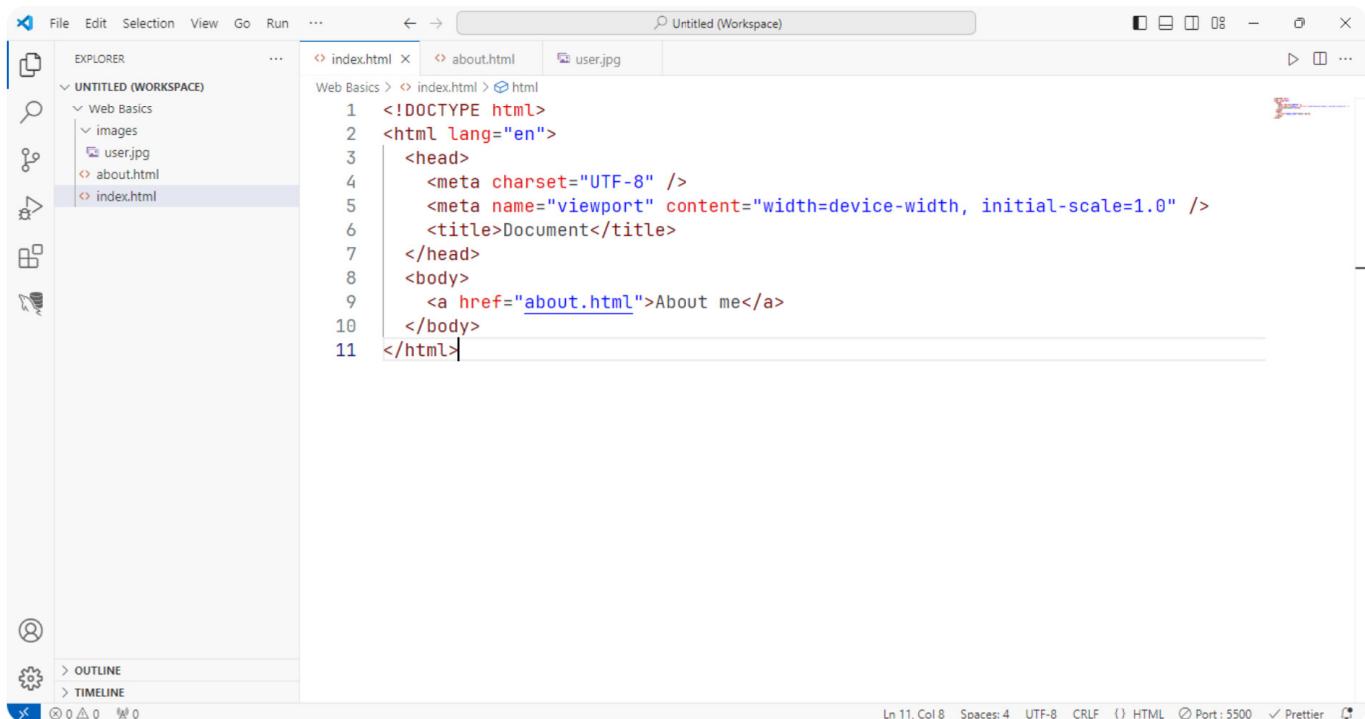
11. Hyperlinks

Hyperlinks in HTML are references that allow users to navigate from one page to another or to specific sections within a page. They are typically created using the anchor tag ().

11.1 Key Points about Hyperlinks and the Anchor Tag

- Anchor Tag:** The anchor tag is represented by the element, which stands for "anchor." This tag is used to define a hyperlink that links to another resource.
- href Attribute:** The most important attribute of the anchor tag is href, which specifies the URL (Uniform Resource Locator) of the page or resource the link points to.

1. Create another .html document in the same working directory as the **index.html**.
2. Let's us name the new .html document as **about.html** to which we want to redirect to.



```

File Edit Selection View Go Run ...
UNTITLED (WORKSPACE)
Web Basics > index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>Document</title>
7  </head>
8  <body>
9      <a href="about.html">About me</a>
10 </body>
11 </html>

```

Ln 11, Col 8 Spaces: 4 UTF-8 CRLF {} HTML Port: 5500 ✓ Prettier

index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Document</title>
</head>
<body>
    <a href="about.html">About me</a>
    <a href="/images/user.jpg">View Image</a>
</body>
</html>

```

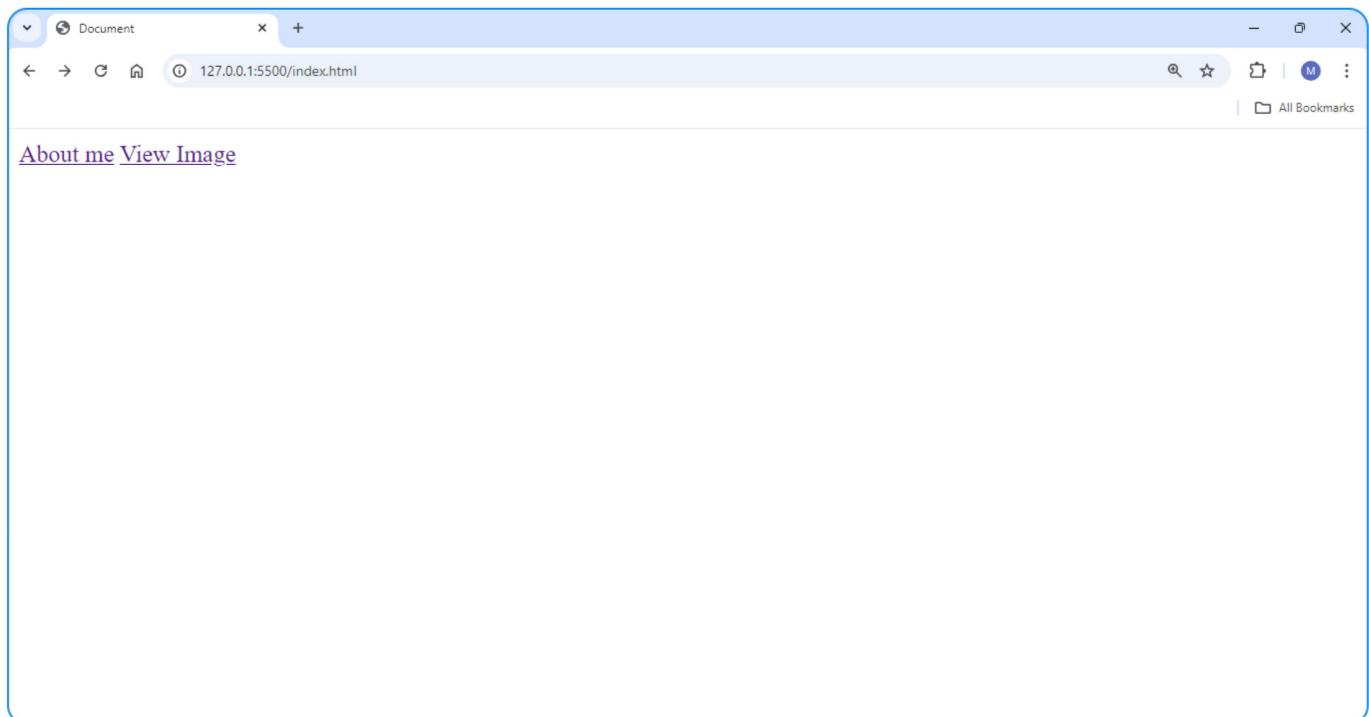
about.html

```

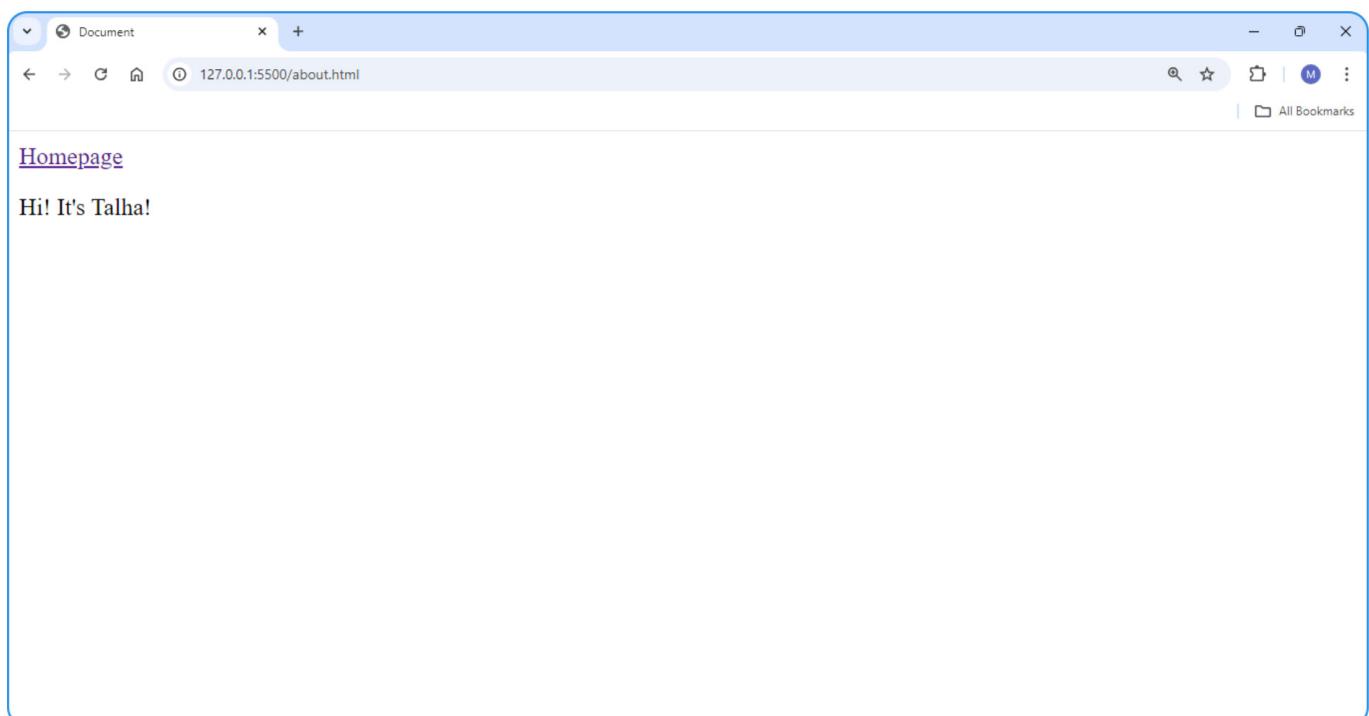
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />

```

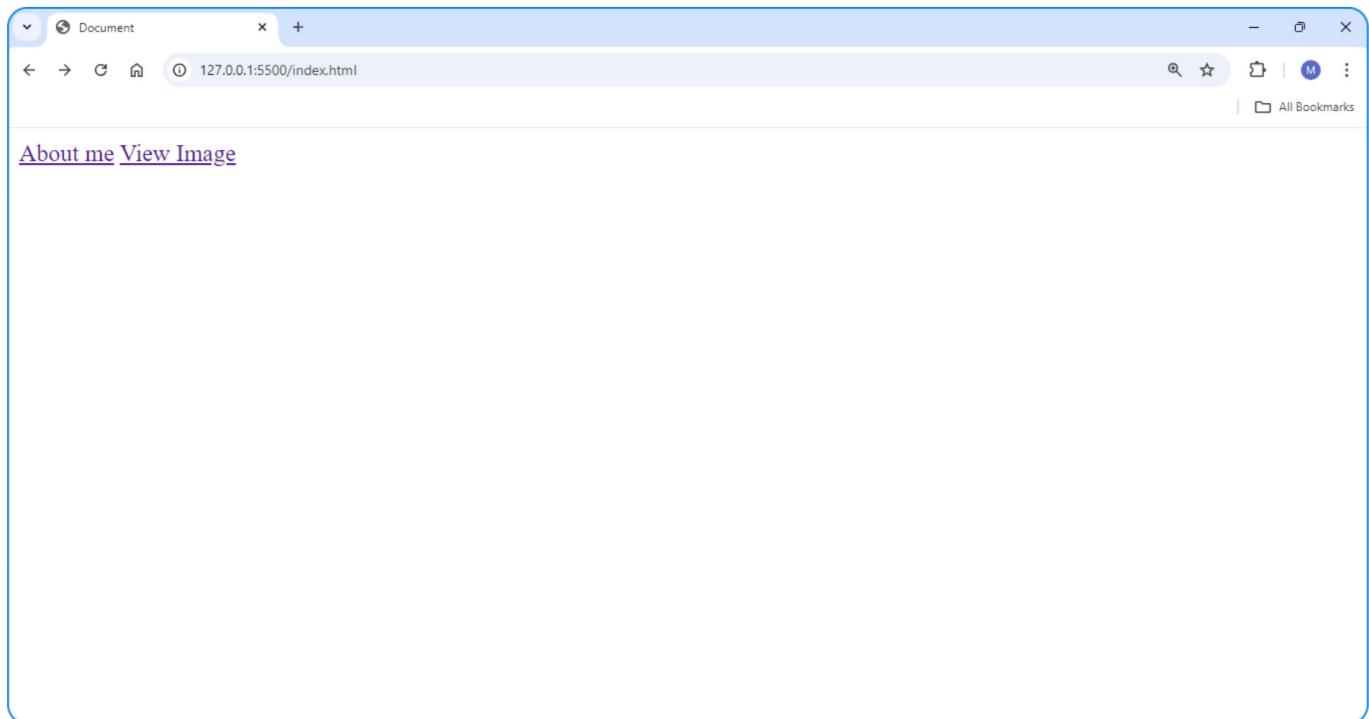
```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Document</title>
</head>
<body>
<a href="index.html">Homepage</a>
<p>Hi! It's Talha!</p>
</body>
</html>
```



3. If you click on **About me**, you will be directed to the **about.html** page.



4. Click on **Homepage** to go to **index.html**.



5. As we also added an anchor for the image, click on **View Image** to view the image.

